Воронежский государственный университет
Факультет прикладной математики, информатики и механики

Математика, информационные технологии, приложения

Межвузовская научная конференция молодых ученых и студентов

Воронеж,
26 апреля 2021 г.

Воронеж
Издательско-полиграфический центр
«Научная книга»
2021

УДК 531(063)+51-7(063)
ББК 22.2я5+22.1я5
М34

Математика, информационные технологии, приложения : сборник трудов Межвузовской научной конференции молодых ученых и студентов, Воронеж, 26 апреля 2021 г. – Воронеж : Научная книга, 2021. – 277 с.
М34
ISBN 978-5-4446-1555-3

Традиционно конференция является междисциплинарной. Важнейшая ее цель – ознакомление молодых специалистов с новыми веяниями в современной науке, а также обмен знаниями и достижениями в предложенных научных направлениях.

УДК 531(063)+51-7(063)
ББК 22.2я5+22.1я5

ISBN 978-5-4446-1555-3

УДК 80/81; 004

ПРОГРАММА ИССЛЕДОВАНИЯ КОРРЕЛЯЦИИ РАЗМЕРОВ СЛОВАРЯ И ТЕКСТА

К. Т. Акчурина

Воронежский государственный университет

Введение

Создание программных инструментов в рамках задачи автоматизации лингвистических исследований имеет важное значение, поскольку для получения результата необходимо обработать значительное количество материала, что вручную весьма проблематично. Разрабатываемое приложение позволяет оценивать и сравнивать богатство словарей авторов с точки зрения коэффициента лексического разнообразия их текстов.

1. Задача исследования словаря и текста

1.1. Постановка задачи

Исходные данные: тексты определенного автора в хронологическом порядке.

1. Каждый текст характеризуется 2-мя параметрами: длиной (в словоупотреблениях, то есть текстовых словах, последовательностях символов между двумя пробелами, если исключить знаки препинания) и размером словаря (количеством разных слов или лемм [1], каждая из которых представляет множество словоформ данного слова; например, СТОЛ=лемма; стола, столы, столом, столов – словоформы).

2. Каждое слово характеризуется в тексте частотой: суммой частот своих словоформ.

3. Задача состоит в том, чтобы проследить рост текстов и рост словаря данного автора нарастающим итогом (по времени создания).

Это значит, что Текст 1 (Т1) и Текст 2 (Т2) имеют длину равную длине Т1 + Длина Т2.

4. Словари текстов суммировать нельзя: это всегда будут пересекающиеся множества. М1 будет содержать слова, представленные только в Т1, а М3 – только слова, представленные в тексте Т2, а вот М2 будет содержать слова, представленные в Т1 и Т2. Эти слова нельзя суммировать, но необходимо суммировать их частоты в Т1 и Т2.

5. Словарь множества Т1+Т2 будет равен сумме множеств М1+М2+М3.

6. В множестве Т1+Т2 все слова также ранжируются по их частоте.

7. Нужно распределение слов по частоте в Т1, Т1+Т2, Т1+Т2+Т3, Т1+Т2+Т3+Т4 и т.д. — до последнего текста данного автора.

1.2. Основные определения

Коэффициент лексического разнообразия (КоЛеР) – величина, отображающая степень богатства словаря текста заданной длины [2] (1).

$$L_d = \frac{N_{lex}}{N}, \tag{1}$$

где L_d – КоЛеР;

N_{lex} – количество уникальных лемм в анализируемом тексте (словарь текста);

N – общее число словоформ в тексте (длина).

Данная характеристика широко используется при исследовании стилистики текстов, сравнении оригинальных и переведенных произведений [3].

Известно, что активный словарь автора – величина конечная. Следовательно, при рассмотрении достаточно большого объема текстов этот прирост уменьшается, а потом и вовсе прекращается. Благодаря разрабатываемому приложению мы сможем не только оценить предельный словарь автора, но и посчитать КоЛеР для каждого текста последовательности, который поможет определить время прекращения роста словаря, что может быть полезно в иных литературных исследованиях. Наряду с коэффициентом лексического разнообразия мы можем составить и другие характеристики текста, а именно *доли новых слов* для каждого текста и *доли покрытия текста новыми словами*, которые также являются полезными характеристиками при исследовании идиолекта отдельно взятого автора.

Идиолект – язык отдельного человека, представленного корпусом всех созданных им текстов.

Доля новых слов – отношение количества новых слов в данном тексте к полному словарю произведения.

Доля покрытия текста новыми словами – отношение количества новых лемм к длине всего текста.

2. Описание приложения

2.1. Средства реализации

Приложение разрабатывается на языке JAVA в силу его кроссплатформенности. Для лемматизации текста используется фреймворк Tawt. В качестве хранилища данных использована документно-ориентированная база данных MongoDB. Реляционная вариант хранения данных в данной задачи не применим.

2.2. Интерфейс пользователя

Разрабатываемое приложение предоставляет интерфейс загрузки данных, представленный на рис. 2.1.

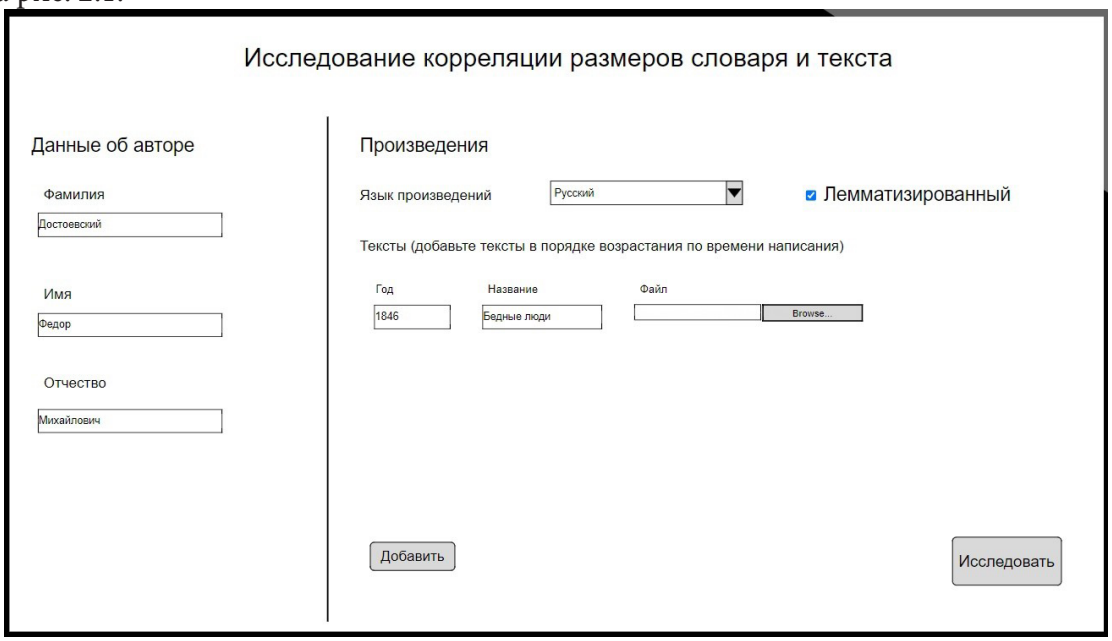


Рис. 2.1. Интерфейс стартовой страницы приложения

После завершения обработки пользователю предоставляется сводная таблица, показывающая словарь каждого текста, количество новых слов, долю новых слов, КоЛеР на каждом этапе (рис 2.2).

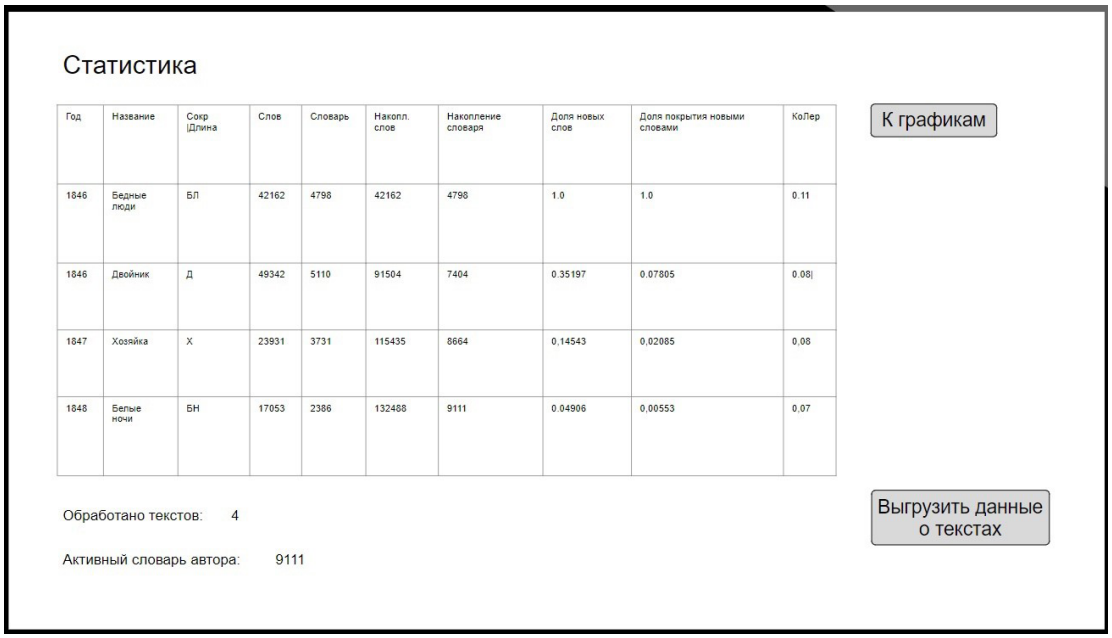


Рис. 2.2. Сводная таблица по загруженным ранее текстам

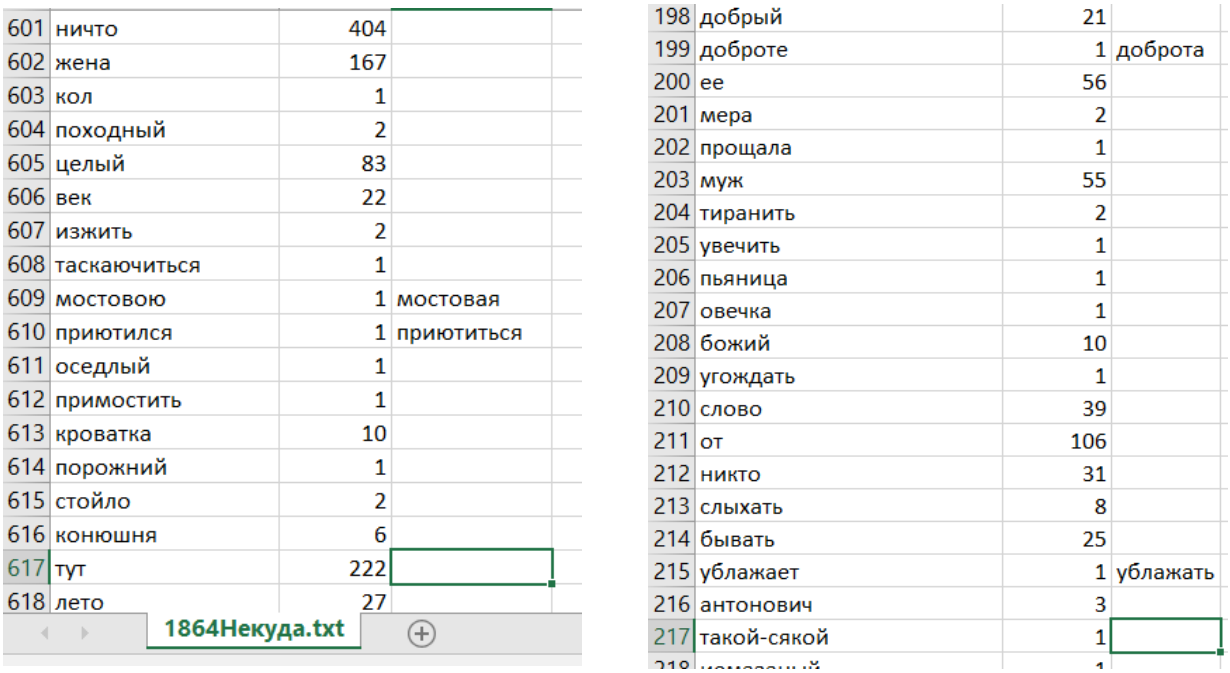
3. Проблемы реализации

Приложение использует библиотеки машинного обучения для привидения слова в начальную форму. Изначально для лемматизации была выбрана библиотека Natasha, реализованная на Python. К сожалению, на больших наборах данных, библиотека выдает нереалистичные результаты за счет накопления погрешностей. Так, при анализе 10 текстов Н. С. Лескова, был получен предельный словарь автора в количестве 39570 слов, что, очевидно, далеко от правды (рис 3.1).

	A	B	C	D	E	F	G	H
1							Накопл.	
2	№	Год	Текст	Сокр	Длина	Слов	Текст	Словарь
3	1	1863	Житие одной бабы	**ЖОБ**	34803	6022	34803	6022
4	2	1864	Некуда	**Нек**	185832	17272	220635	19454
5	3	1865	Обойденные	**Обо**	81437	9663	302072	22717
6	4	1866	Воительница	**Вои**	23501	4198	325573	23886
7	5	1866	Островитяне	**Ост**	54296	8026	379869	26035
8	6	1869	Старые годы в селе Плодомасове	**СГП**	25678	5295	405547	27360
9	7	1870	На ножах	**НаН**	234509	17737	640056	33911
10	8	1871	Смех и горе	**СиГ**	55242	8603	695298	35819
11	9	1872	Загадочный человек	**ЗЧе**	31381	5579	726679	36706
12	10	1872	Соборяне	**Соб**	92810	12221	819489	39570

Рис. 3.2. Словари произведений Лескова, составленные программой

Полученные на данном этапе результраты стали основанием для принятия решения о замене библиотеки. Была выбран фреймворк Tawt. Производительность оказалась значительно выше, но результаты все еще нереалистичны. На том же наборе данных был получен активный словарь автора, равный 33420 словам (рис. 3.3).



601	ничто	404	
602	жена	167	
603	кол	1	
604	походный	2	
605	целый	83	
606	век	22	
607	изжить	2	
608	таскаючиться	1	
609	мостовою	1	мостовая
610	приютился	1	приютиться
611	оседлый	1	
612	примостить	1	
613	кроватка	10	
614	порожний	1	
615	стойло	2	
616	конюшня	6	
617	тут	222	
618	лето	27	
198	добрый	21	
199	доброте	1	доброта
200	ее	56	
201	мера	2	
202	прощала	1	
203	муж	55	
204	тиранить	2	
205	увечить	1	
206	пьяница	1	
207	овечка	1	
208	божий	10	
209	угождать	1	
210	слово	39	
211	от	106	
212	никто	31	
213	слыхать	8	
214	бывать	25	
215	ублажает	1	ублажать
216	антонович	3	
217	такой-сякой	1	

Рис. 3.3. Сводная таблица по текстам Лесоква, составленная с помощью библиотеки ***Tawt***

На основании вышеизложенного было принято решение позволить пользователю загружать уже лемматизированные тексты.

Заключение

На данный момент программные средства решают задачу только для русскоязычных текстов. Далее планируется расширить функциональность приложения, чтобы появилась возможность анализа текстов на разных языках. Также необходимо уделить особое внимание решению задачи лемматизации для получения более точных результатов.

Литература

1. *Белогонов, Г. Г.* Компьютерная лингвистика и перспективные информационные технологии : теория и практика построения систем автомат. обраб. текстовой информ. / Г. Г. Белоногов, Ю. П. Калинин, А. А. Хорошилов. – М. : Рус. мир, 2004 (Информационно-издат. агентство Русский мир). – 246 с.

2. *Кретов, А. А.* О некоторых количественных характеристиках фрактальности в языке / А. А. Кретов, М. В. Половинкина, И. П. Половинкин, М. В. Ломец // Информатика: проблемы, методы, технологии: сборник материалов XX международной научно-методической конференции / под редакцией А. А. Зацаринного, Д. Н. Борисова; Воронеж, Воронежский государственный университет, 13-14 февраля 2020 г. – Воронеж: Издательство «Научно-исследовательские публикации» (ООО «Вэлборн»), 2020, С. 1627–1634.

3. Коэффициент лексического разнообразия. – Режим доступа: https://ru.wikipedia.org/wiki/Коэффициент_лексического_разнообразия – (Дата обращения: 14.04.2021).

Акчурина Карина Тагировна – студентка 3-го курса кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: akchurina.kar@yandex.ru

Воронина Ирина Евгеньевна (научный руководитель) – д-р физ.-мат. наук, проф., профессор кафедры Программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: irina.voronina@gmail.com

УДК 004.023

РЕШЕНИЕ ЗАДАЧИ ПЛАНИРОВАНИЯ ОПТИМАЛЬНОГО ГРАФИКА РЕМОНТОВ НЕФТЯНЫХ СКВАЖИН С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА РОЯ ЧАСТИЦ

С. А. Анохин

Воронежский государственный университет

Введение

В процессе численного решения многих задач математического моделирования необходимо находить точку экстремума некоторой функции многих переменных (целевой функции).

Традиционные методы локальной оптимизации обычно не позволяют решать подобные задачи, т. к. с их помощью, как правило, удается найти только один локальный экстремум. Во многих областях для получения близкого к оптимальному решения за время, допустимое для функционирования системы, успешно применяются стохастические эвристические методы оптимизации, такие как алгоритм имитации отжига, эволюционные и роевые алгоритмы. Они относятся к методам локального поиска и основаны на идеях, взятых из природы.

В работе предложена модификация алгоритма роя частиц, позволяющая решать задачу планирования оптимального расписания ремонтов нефтяных скважин несколькими ремонтными бригадами.

1. Постановка задачи

В данной статье рассматривается следующая задача планирования: имеются скважины, добывающие нефть, со временем производительность скважины уменьшается и возникает необходимость ее ремонта. Ремонтные работы проводят специальные бригады.

По каждой скважине имеется следующая информация:

- производительность скважины **ДО** ремонта;
- производительность скважины **ПОСЛЕ** ремонта;
- длительность ремонта (во время ремонта скважина не добывает нефть).

Для ремонтов выделяется определенное количество дней – ремонтный период (обозначим RP). Эффективность полученного расписания (плана) исчисляется объемом добытой нефти за полный период (обозначим FP).

Необходимо составить расписание ремонтов, обеспечивающее максимальную добычу нефти.

2. Математическая модель

2.1. Эффективность скважины

Пусть имеется скважина (обозначим W) с параметрами:

- p_b – дебет нефти до ремонта;
- p_a – дебет нефти после ремонта;
- dur – длительность ремонта.

Объем нефти, добытой скважиной W можно вычислить по формуле:

$$Z(W) = p_b * d_b + p_a * d_a, \quad (1)$$

где d_b – количество дней работы скважины до ремонта,

d_a – количество дней работы скважины после ремонта.

2.2. Модификация задачи о назначениях

Если рассмотреть поставленную задачу как задачу поиска максимально эффективного соответствия «номер ремонта – номер скважины» для одной бригады, тогда задача приобретает черты задачи о назначениях.

Введем переменные следующим образом:

x_{ij} – бригада выполняет i-й ремонт на j-й скважине. Целевая функция с ограничениями может быть описана следующим образом:

$$\max S = \sum_{i=1}^{n}\sum_{j=1}^{n} Z(W_j) * x_{ij} \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n \tag{4}$$

$$x_{ij} \in \{0,1\}, \quad i = 1,2,\ldots,n, \quad j = 1,2,\ldots,n \tag{5}$$

Кроме того, в задаче планирования графика ремонтных работ необходимо учитывать ограничение на количество скважин, которые могут быть отремонтированы за ремонтный период (RP).

$$\sum d_j \le RP, \tag{6}$$

где d_j – длительность ремонта j-й скважины.

2.3. Переход к задаче с переменным количеством ремонтных бригад

Описанная выше модель определяет очередь скважины в ремонтном плане одной бригады. Чтобы получить ремонтный план, который распределит ремонтные работы среди некоторого числа бригад (обозначим n – количество бригад) предлагается переход к ремонтному плану следующего вида:

$$[X_1] \ [X_1] \ \ldots \ [X_n], \tag{7}$$

где $[X_k]$ – матрица $N \times N$ – ремонтный план k-й бригады, где N – количество скважин, подлежащих ремонту.

3. Алгоритм роя частиц

3.1. Суть метода

В основе алгоритма лежит модель, описывающая поведение индивидов в толпе или особей в стае. При нахождении скорости движения частицы учитывается ее скорость на предыдущих итерациях, направление от текущего положения частицы к ее положению с наилучшим значением целевой функции за все время поиска, а также координаты лучшего решения (лидера) за все время поиска [3].

Наибольшей сложностью в применении роевых алгоритмов является их настройка и доработка для различных видов оптимизационных задач, подбор значений коэффициентов алгоритмов для получения высокой эффективности на различных классах задач. Для различных задач оптимизации лучше подходят различные роевые алгоритмы, но методик их выбора не существует [2].

Введем следующие обозначения:

X – матрица текущего приближенного решения(плана);

V – матрица скоростей;

B – матрица лучшего решения агента;

G – матрица лучшего глобального решения.

Все матрицы имеют размерность $N \times N$, где N – количество скважин, претендующих на ремонт.

Алгоритм роя частиц может быть описан следующей последовательностью шагов.

1. Инициализация матриц X, $B = X$ для каждого агента.
2. Поиск G среди всех агентов.
3. Инициализация V для каждого агента.
4. Вычисление X, V по итерационным формулам.
5. Обновление B и G.

Шаги 4, 5 повторяются пока не выполнится критерий останова (например, лучший глобальный результат не изменяется в течении K итераций).

Рассмотрим детально каждый шаг.

Инициализация матрицы X выполняется путем случайного построения матрицы, удовлетворяющей критериям (3)–(5). На этом этапе решение В совпадает с решением X.

После инициализации матриц X всех агентов происходит поиск G – лучшего решения среди всех инициализированных.

Затем инициализируются матрицы скоростей каждого агента V:

$$V = \varphi_1(B - X) + \varphi_2(G - X). \tag{8}$$

Для определения значений φ_1 и φ_2 используется следующее правило:

Если $B != X$ и $G = X$, то $\varphi_1 = 1;\ \varphi_2 = 0$.

Если $B = X$ и $G != X$, то $\varphi_1 = 0;\ \varphi_2 = 1$.

Иначе φ_1 выбирается случайным образом в пределах $[0, 0.2]$, $\varphi_2 = 1 - \varphi_1$.

На шаге 4 вычисление матриц X_{i+1} и V_{i+1} происходит по следующим формулам:

$$V_{i+1} = \lambda_1 V_i + \lambda_2(\varphi_1(B_i - X_i) + \varphi_2(G_i - X_i)) \tag{9}$$

$$X_{i+1} = X_i + V_{i+1}. \tag{10}$$

Значения φ_1 и φ_2 определяются, руководствуясь ранее описанным правилом. Значения λ_1 и λ_2 определяются по следующему правилу:

Если $B_i = X_i$ и $G_i = X_i$, то $\lambda_1 = 1,\ \lambda_2 = 0$.

Иначе λ_1 выбирается случайным образом в пределах $[0.8, 1]$, $\lambda_2 = 1 - \lambda_1$.

3.2. Особенности в рамках задачи

После применения формул (9)–(10) матрица X_{i+1} может содержать действительные значения. В случае наличия отрицательных значений необходимо применить алгоритм восстановления к каждому отрицательному элементу матрицы.

Алгоритм 1.

Вход. Матрица X, row – индекс строки отрицательного элемента, col – индекс столбца отрицательного элемента.

1. $val = X_{row,col}$ – значение отрицательного элемента, к которому применяется оператор.
2. Поиск $X_{l,col}$ – максимального элемента в столбце col.
3. Вычисление u – количества положительных элементов в строке row.
4. $X_{row,col} = 0$.
5. Пересчет $X_{l,col} = X_{l,col} - |val|$.
6. Пересчет элементов в строке row: $X_{row,j} = X_{row,j} - \dfrac{|val|}{u}$ для всех $X_{row,j} > 0$.
7. Пересчитываем элементы в строке l: $X_{l,j} = X_{l,j} + \dfrac{|val|}{u}$ для всех $X_{l,j} > 0$.

Алгоритм применяется пока в матрице существуют отрицательные элементы. После применения алгоритма, описанного выше, сохраняется вероятность наличия в матрице вещественных положительных значений. Так как эффективность плана можно посчитать только для матрицы, у которой $x_{ij} \in \{0,1\} \quad \forall i, j,$ для приведения матрицы X к такому виду предлагается следующий алгоритм:

Алгоритм 2.

1. Инициализация матрицы X' таких же размеров.
2. Для каждой строки row матрицы X выполнить:
   a. поиск индекса col – максимального элемента в строке.
   b. $X'[row][col] = 1.$

После применения данного алгоритма матрица может перестать соответствовать ограничению (9), а именно:

$$\exists j : \sum_{i=1}^{n} x_{ij} \neq 1 \tag{11}$$

Такую матрицу необходимо привести к виду, который будет удовлетворять ограничениям.

Рассмотри следующий пример.

Допустим, в результате вычислений получен график ремонтных работ одной из бригад:

$$X = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Он является недопустимым, так как 2-й и 4-й ремонты проводятся на одной скважине.

Для исправления плана предлагается следующий алгоритм:

1. Определение столбцов (строки не могут быть нарушены), в которых нарушены ограничения **brokenCols**
2. Разделение колонок на 2 группы: **addCols** – для добавления значения, **subCols** – для вычитания.
3. Сортировка **addCols** по «эффективности скважины» по убыванию.
4. Сортировка **subCols** по возрастанию по номеру строки row, начиная с которой сумма первых row значений в столбце >1
5. Для каждого столбца **col** из **subCols**:
   Пока сумма столбца >1 повторять:
   i. Найти строку **row**, начиная с которой сумма первых **row** значений в столбце >1.
   ii. Посчитать **v** = min(M[row][col], сумма столбца–1)
   iii. M[row][col] = M[row][col] – **v**
   iv. Для всех столбцов **c** из **addCols**:
      1) найти **d** = min(**v**, 1–сумма столбца **c**)
      2) M[row][c] = **d**
      3) v=v–d

Кроме, описанных выше, проблем, которые возникают при построении графика ремонтов для каждой бригады в отдельности, также возникает и проблема устранения конфликтных ситуаций между разными бригадами.

Рассмотрим пример.

Допустим, на каком-то этапе алгоритма был получен следующий график ремонтов:

$$\text{Бригада 1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad \text{Бригада 2} = X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Обе бригады хотя выполнить второй ремонт на скважине № 3, что делает наш план недопустимым. Для устранения подобных конфликтов предлагается следующий алгоритм, который выполняется для каждой строки матриц X_i:

Алгоритм 3.

1. В каждой строке ищем конфликтующие бригады.

2. Из конфликтующих выбираем бригаду, которая освободится раньше всех. Матрицу X этой бригады оставляем без изменений.

3. В матрицах остальных бригад необходимо выполнить обмен текущей строки i с первой строкой k, которая:

a. $k > i$;

b. не участвовала в обменах ранее.

Заключение

В работе предложен подход к применению алгоритмов роевого интеллекта для решения оптимизационных задач планирования графика ремонтных работ.

Представлен алгоритм, разработанный на основе алгоритма роя частиц, и создана его программная реализация.

Литература

1. *Мальковский С. И., Пересветов В. В.* Решение нелинейных транспортных задач методом роя частиц // Информатика и системы управления. – ВЦ ДВО РАН, Хабаровск. – 2012. – С. 54–64.

2. *Матренин П. В.* Разработка адаптивных алгоритмов роевого интеллекта в проектировании и управлении техническими системами : дис. … канд. техн. наук ; Новосибирский государственный технический университет. – 2018. – 197 с.

3. *Королев С. А., Майков Д. В.* Модификация алгоритма роя частиц на основе метода анализа иерархий // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2019. – № 4. – С 36–46.

4. *Трегубов А. Г., Медведев С. Н.* Вероятностный подход к решению трехиндексной аксиальной задачи о назначениях. вычислительный эксперимент // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2015. – № 4. – С. 31–37.

5. *Павленко А. И.* Сравнительный анализ модифицированных методов муравьиных колоний. – Московский авиационный институт. – 2012. – 13 с.

6. *Chi-Jen Lin and Ue-Pyang Wen*. A Labeling algorithm for the fuzzy assignment problem // Fuzzy sets and systems. – 2004. – 142.– P. 373–379.

Анохин Сергей Александрович – студент 2-го курса магистратуры кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: sergeyanokhin2014@gmail.com

Каплиева Наталья Алексеевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: kaplieva@amm.vsu.ru

УДК 517.972

ЧИСЛЕННОЕ НАХОЖДЕНИЕ МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ, СОДЕРЖАЩИХ СТОХАСТИЧЕСКИЕ ПРОЦЕССЫ. МОДЕЛЬ БОЕВЫХ ДЕЙСТВИЙ ЛАНЧЕСТЕРА

И. И. Бакаев

Воронежский государственный университет

Введение

В общем виде модель боевых действий можно описать системой:

$$\begin{cases} \dot{x} = ax + bx + cy + d \\ \dot{y} = ey + fyx + gx + h \end{cases}$$

Коэффициенты a и e характеризуют скорость небоевых потерь, b и f характеризуют скорость потерь из-за воздействия по площадным целям, d и h характеризуют подходящие или отходящие резервы.

Часто в модели имеются только коэффициенты b и f. В этом случае количество жертв пропорционально количеству встреч между противоборствующими сторонами. Наиболее актуально подобное взаимодействие тогда, когда две стороны располагаются на общей территории.

Наибольшее применение модель Ланчестера снискала в форме:

$$\begin{cases} \dot{x} = -ax - cy \pm d \\ \dot{y} = -ey - gx \pm h \end{cases}$$

1. Математическая модель со случайными коэффициентами

В реальных условиях коэффициенты определяются приближённо и зависят от внешних факторов. Их можно моделировать как случайные процессы.

Мы рассматриваем задачу:

$$\begin{cases} \ddot{x} = -f(t)\dot{x} - \varepsilon(t)x \\ x(t_0) = x_0 \\ \dot{x}(t_0) = x_1 \end{cases}$$

Обозначим данную задачу (1).

Здесь $f(t)$ – случайный процесс, $\varepsilon(t)$ – заданная функция.

Пусть нам известны: $\mathrm{M}[x_0]$, $\mathrm{M}[x_1]$.

Будем также считать, что известен характеристический функционал:

$$\varphi_f(u) = \mathrm{M}[e^{i\int_{t_0}^{t_1} f(s)u(s)ds}].$$

Характеристический функционал – преобразование Фурье плотности распределения случайного процесса. [1]

Задача состоит в нахождении математического ожидания решения.

2. Вспомогательная детерминированная модель

Введём обозначение:

$$\mathrm{W} = e^{i\int_{t_0}^{t_1} f(s)u(s)ds}.$$

Умножим $\ddot{x} = f(t)\dot{x} - \varepsilon(t)x$ на W и возьмём математическое ожидание:

$$\mathrm{M}[\ddot{x}\mathrm{W}] + \mathrm{M}[f(t)\dot{x}\mathrm{W}] + \mathrm{M}[\varepsilon(t)x\mathrm{W}] = 0.$$

Пусть $y(t,u) = \mathrm{M}[x(t)\mathrm{W}]$.

Заметим, что $y(t,0) = \mathrm{M}[x(t)]$.

Также заметим, что:

$$\mathrm{M}[\ddot{x}\mathrm{W}] = \frac{\partial^2 y(t,u)}{\partial t^2},\ \mathrm{M}[f(t)\dot{x}\mathrm{W}] = \frac{1}{i}\frac{\delta}{\delta u(t)}\frac{\partial y(t,u)}{\partial t}.$$

Здесь $\dfrac{\delta}{\delta u(t)}$ – вариационная производная [2].

Найдём теперь $\dfrac{\delta\varphi_f}{\delta u(t)}$ и $\dfrac{\delta y(t,u)}{\delta u(t)}$:

$$\frac{\delta\varphi_f}{\delta u(t)} = \mathrm{M}[if(t)\mathrm{W}],\ \frac{\delta y(t,u)}{\delta u(t)} = \mathrm{M}[x(t)\mathrm{W}if(t)].$$

Получаем:

$$\frac{\partial^2 y}{\partial t^2} + \frac{1}{i}\frac{\delta}{\delta u(t)}\frac{\partial y}{\partial t} + \varepsilon(t)y = 0.$$

Умножим $x(t_0) = x_0$ и $\dot{x}(t_0) = x_0$ на W и возьмём математическое ожидание:

$$\mathrm{M}[x(t_0)\mathrm{W}] = \mathrm{M}[x_0\mathrm{W}] = \mathrm{M}[x_0]\mathrm{M}[\mathrm{W}] = \mathrm{M}[x_0]\varphi_f(u),$$

$$\mathrm{M}[\dot{x}\mathrm{W}] = \mathrm{M}[x_1]\mathrm{M}[\mathrm{W}] = \mathrm{M}[x_1]\varphi_f(u),$$

$$y(t_0,u) = \mathrm{M}[x_0]\varphi_f(u),$$

$$\frac{\partial y(t,u)}{\partial t} = \mathrm{M}[x_1]\varphi_f(u).$$

Имеем задачу:

$$\begin{cases} \dfrac{\partial^2 y}{\partial t^2} + \dfrac{1}{i}\dfrac{\delta}{\delta u(t)}\dfrac{\partial y}{\partial t} + \mathring{a}(t)y = 0 \qquad (*) \\ y(t_0,u) = \mathrm{M}[x_0]\varphi_f(u) \\ \dfrac{\partial y(t,u)}{\partial t} = \mathrm{M}[x_1]\varphi_f(u) \end{cases}$$

В уравнении (*) имеется комплексный коэффициент i^{-1}. Иногда это приводит к усложнению расчётов. Избавимся от i^{-1}. С этой целью введём:

$$Y(t,\xi) = y(t,i\xi),\ \text{где } \xi \text{ – вещественная переменная.}$$

Получаем:

$$\frac{\partial^2 Y(t,\xi)}{\partial t^2} = \frac{\partial^2 y(t,i\xi)}{\partial t^2}$$

$$\frac{\partial Y(t,\xi)}{\partial t} = \frac{\partial y(t,i\xi)}{\partial t}$$

$$\frac{\delta}{\delta\xi(t)}\frac{\partial Y(t,\xi)}{\partial t} = \frac{\delta}{\delta\xi(t)}\frac{\partial y(t,i\xi)}{\partial t} = i\frac{\delta}{\delta u}\frac{\partial y(t,u)}{\partial t},\ \text{при } u = i\xi.$$

Таким образом уравнение (*) записывается в виде:

$$\frac{\partial^2 Y(t,\xi)}{\partial t^2} - \frac{\delta}{\delta\xi(t)}\frac{\partial Y(t,\xi)}{\partial t} + \varepsilon(t)Y(t,\xi) = 0 \qquad (**)$$

3. Численное нахождение математического ожидания решения

Заметим, что $Y(t,0) = y(t,0) = \mathrm{M}[x(t)]$.

Пусть $h_t(s) = \begin{cases} 1, \text{ при } s \in [t, t+\gamma] \\ 0, \text{ иначе} \end{cases}$, где γ – вещественное число > 0.

Тогда:

$$\int_{t_0}^{t_1} A(s)h_t(s)ds = \int_{t}^{t+\gamma} A(s)ds \approx A(t)\gamma$$

$$\frac{\delta Y(x)}{\delta u(t)} \approx \frac{Y(x+h_t) - Y(x)}{\gamma}$$

Пусть теперь:

$$t_i = t_0 + i\tau$$

$$u_k(s) = kh_t(s)$$

$$Y_k^i = Y(t_i, u_k)$$

$$\varepsilon_i = \varepsilon(t_i)$$

Получаем:

$$\frac{\partial Y}{\partial t} \approx \frac{Y(t+\tau) - Y(t)}{\tau}$$

$$\frac{\partial Y_k^i}{\partial t} \approx \frac{Y_k^{i+1} - Y_k^i}{\tau}$$

$$\frac{\partial^2 Y}{\partial t^2} \approx \frac{Y_k^{i+2} - 2Y_k^{i+1} + Y_k^i}{\tau^2}$$

$$\frac{\delta Y}{\delta u(t)} \approx \frac{Y_{k+1}^i - Y_k^i}{\gamma}$$

$$\frac{\delta}{\delta u(t)}\frac{\partial Y}{\partial t} \approx \frac{Y_{k+1}^{i+1} - Y_{k+1}^i - Y_k^{i+1} + Y_k^i}{\gamma\tau}$$

Подставим данные выражения в уравнение (**) и получим разностное уравнение:

$$\frac{Y_k^{i+2} - 2Y_k^{i+1} + Y_k^i}{\tau^2} - \frac{Y_{k+1}^{i+1} - Y_{k+1}^i - Y_k^{i+1} + Y_k^i}{\tau\gamma} + \varepsilon^i Y_k^i = 0.$$

Выразим отсюда Y_k^{i+2}:

$$Y_k^{i+2} = 2Y_k^{i+1} - Y_k^i + \frac{\tau(Y_{k+1}^{i+1} - Y_{k+1}^i - Y_k^{i+1} + Y_k^i)}{\gamma} - \tau\varepsilon^i Y_k^i.$$

4. Алгоритм численного нахождения математического ожидания решения задачи (1)

1. Задаём количество точек N, шаг по времени τ, γ, $\mathrm{M}[x_0]$, $\mathrm{M}[x_1]$, $a(s)$, $b(s1, s2)$, $\varphi_f(u)$
2. Пускаем цикл по k $(0 \le k \le N-1)$
3. Считаем $Y_k^0 = \mathrm{M}[x_0]\varphi_f(\sqrt{-1}kh_{t_0})$
4. Печатаем Y_0^0
5. Пускаем цикл по k $(0 \le k \le N-1)$
6. Считаем $Y_k^1 = Y_k^0 + \tau\mathrm{M}[x_1]\varphi_f(\sqrt{-1}kh_{t_0})$
7. Печатаем Y_0^1

8. Пускаем цикл по i $(0 \le i \le N-2)$

9. Пускаем вложенный цикл по k $(0 \le k \le N-i-1)$

10. Считаем $Y_k^{i+2} = 2Y_k^{i+1} - Y_k^i + \dfrac{\tau(Y_{k+1}^{i+1} - Y_{k+1}^i - Y_k^{i+1} + Y_k^i)}{\gamma} - \tau\varepsilon^i Y_k^i$

11. Печатаем Y_0^{i+2}

Заключение

В ходе исследования мы получили разностное уравнение для численного нахождения математического ожидания решения дифференциального уравнения, содержащего стохастический процесс.

Благодарности

Выражаю благодарность своему научному руководителю, д-ру физ.-мат. наук, проф., заведующему кафедрой системного анализа и управления Воронежского государственного университета, Задорожнему Владимиру Григорьевичу за помощь во всех аспектах исследования и бесценный вклад в написание данной статьи.

Литература

1. *Задорожний, В. Г.* Дифференциальные уравнения со случайными коэффициентами : учебное пособие для вузов / В. Г. Задорожний; Воронежский государственный университет. – Воронеж : Издательско-полиграфический центр Воронежского государственного университета, 2012. – 98 с.

2. *Задорожний, В. Г.* Методы вариационного анализа : учебное пособие для вузов / В.Г. Задорожний. – М. – Ижевск : НИЦ «Регулярная и хаотическая динамика», Институт компьютерных исследований, 2006. – 316 с.

Бакаев Илья Игоревич – студент 4-го курса кафедры системного анализа и управления Воронежского государственного университета. E-mail: ibakaev1999@gmail.com

Задорожний Владимир Григорьевич (научный руководитель) – д-р физ.-мат. наук, профессор, заведующий кафедрой системного анализа и управления Воронежского государственного университета. E-mail: zador@amm.vsu.ru

УДК 004.412

ИНТЕГРАЦИЯ SPRING BOOT ПРИЛОЖЕНИЯ С PROMETHEUS И GRAFANA

О. А. Безрукова

Воронежский государственный университет

Введение

Большинство проектов, которые содержат различные микросервисы и не только, сталкиваются с проблемой, когда нет возможности отследить какие ресурсы потребляет приложение, в какие моменты времени происходит перезапуск и по какой причине, сколько запросов отправляется и какова продолжительность каждого из них. Данные вопросы помогает решить интеграция продукта с системным монитором, который с помощью сбора метрик позволяет отслеживать заданные параметры и выводить их в виде таблиц и графиков.

1. Используемые инструменты

Prometheus – это бесплатное программное приложение, используемое для мониторинга событий и оповещения. Он поддерживает четыре типа метрик:

• **счётчик** (counter) – хранит значения, которые увеличиваются с течением времени (например, количество запросов к серверу);

• **шкала** (gauge) – хранит значения, которые с течением времени могут как увеличиваться, так и уменьшаться (например, объём используемой оперативной памяти или количество операций ввода-вывода);

• **гистограмма** (histogram) – хранит информацию об изменении некоторого параметра в течение определённого промежутка (например, общее количество запросов к серверу в период с 11 до 12 часов и количество запросов к этому же серверу в период с 11.30 до 11.40);

• **сводка результатов** (summary) – как и гистограмма, хранит информацию об изменении значения некоторого параметра за временной интервал, но также позволяет рассчитывать квантили для скользящих временных интервалов.

Grafana – это платформа с открытым исходным кодом для визуализации, мониторинга и анализа данных. Она позволяет пользователям создавать dashboard с панелями, каждая из которых отображает определенные показатели в течение установленного периода времени. Каждый dashboard универсален, поэтому его можно настроить для конкретного проекта или с учетом любых потребностей разработки и/или бизнеса.

2. Интеграция Spring Boot приложения с Prometheus и Grafana

В данном разделе описана проблема, поставлена задача, а также предложен один из возможных способов ее решения.

2.1. Проблема

Невозможность отслеживания состояния и формирования статистики работы web-сервиса или любого другого приложения, которые необходимы для последующего улучшения и оптимизации работы конечного продукта.

2.2. Постановка задачи

Необходимо реализовать интеграцию готового продукта с Prometheus и Grafana, создать пользовательские метрики и вывести их значения в dashboard Grafana. На рис. 1 представлена взаимосвязь сервисов между собой, которую необходимо добавить в имеющееся приложение.

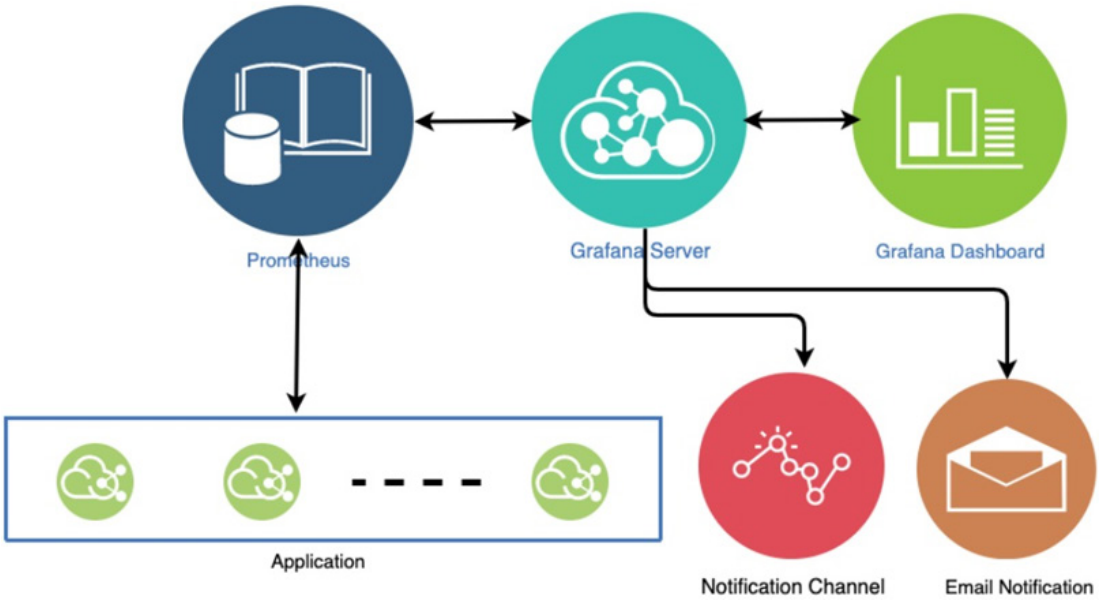


Рис. 1. Взаимодействие Prometheus и Grafana с приложением

2.3. Реализация

Для начала следует добавить библиотеки micrometer-registry-prometheus и spring-boot-starter-actuator в уже существующий проект. Также необходимо добавить в файл конфигурации конечные точки для Prometheus, указав название приложения для мониторинга, детали, для выводимых метрик и то, что необходимы метрики Prometheus.

После этого можно перейти по адресу localhost:8080/actuator/prometheus, где можно увидеть все собранные на данный момент времени метрики приложения.

localhost:8080/actuator/prometheus

```
# HELP process_uptime_seconds The uptime of the Java virtual machine
# TYPE process_uptime_seconds gauge
process_uptime_seconds{application="MonitoringSpringDemoProject",} 24.065
# HELP tomcat_sessions_rejected_sessions_total
# TYPE tomcat_sessions_rejected_sessions_total counter
tomcat_sessions_rejected_sessions_total{application="MonitoringSpringDemoProject",} 0.0
# HELP jvm_classes_unloaded_classes_total The total number of classes unloaded since the Java virtual machine has started execution
# TYPE jvm_classes_unloaded_classes_total counter
jvm_classes_unloaded_classes_total{application="MonitoringSpringDemoProject",} 0.0
# HELP jvm_buffer_memory_used_bytes An estimate of the memory that the Java virtual machine is using for this buffer pool
# TYPE jvm_buffer_memory_used_bytes gauge
jvm_buffer_memory_used_bytes{application="MonitoringSpringDemoProject",id="mapped",} 0.0
jvm_buffer_memory_used_bytes{application="MonitoringSpringDemoProject",id="direct",} 24576.0
# HELP jvm_gc_memory_promoted_bytes_total Count of positive increases in the size of the old generation memory pool before GC to after GC
# TYPE jvm_gc_memory_promoted_bytes_total counter
jvm_gc_memory_promoted_bytes_total{application="MonitoringSpringDemoProject",} 1112752.0
# HELP system_cpu_usage The "recent cpu usage" for the whole system
# TYPE system_cpu_usage gauge
system_cpu_usage{application="MonitoringSpringDemoProject",} 1.0
# HELP jvm_memory_used_bytes The amount of used memory
# TYPE jvm_memory_used_bytes gauge
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="nonheap",id="Compressed Class Space",} 4956400.0
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="nonheap",id="CodeHeap 'non-nmethods'",} 1212672.0
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="nonheap",id="Metaspace",} 3.9196416E7
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="heap",id="G1 Survivor Space",} 8388608.0
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="heap",id="G1 Old Gen",} 5030944.0
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="heap",id="G1 Eden Space",} 2.8311552E7
jvm_memory_used_bytes{application="MonitoringSpringDemoProject",area="nonheap",id="CodeHeap 'non-profiled nmethods'",} 6365824.0
```

Рис. 2. Собранные метрики Prometheus

Однако предоставленные метрики не всегда в полной мере удовлетворяют потребностям, поэтому также представляется возможным создать свои собственные метрики. Например, необходимо задать некоторый счетчик и датчик, который будет получать случайное число в заданном диапазоне. Для этого нужно добавить два дополнительных класса – один будет включать в себя указанные метрики и логику по их изменению, а другой будет обновлять их через определенное время.

Важной особенностью является то, что для получения возможности отслеживать пользовательские метрики, необходимо импортировать MeterRegistry – класс из библиотеки Micrometer. Это дает возможность использовать счетчики, датчики, таймеры и многое другое.

```
# HELP jvm_threads_live_threads The current number of live threads including both daemon and non-daemon threads
# TYPE jvm_threads_live_threads gauge
jvm_threads_live_threads{application="MonitoringSpringDemoProject",} 29.0
# HELP demo_counter_total
# TYPE demo_counter_total counter
demo_counter_total{application="MonitoringSpringDemoProject",} 44.0
# HELP demo_gauge
# TYPE demo_gauge gauge
demo_gauge{application="MonitoringSpringDemoProject",} 3.0
```

Рис. 3. Пользовательские метрики

Следующим шагом является настройка конфигурации для поднятия Prometheus. Для этого используют конфигурационный файл prometheus.yml, где определяют цели – Java-приложение и Prometheus. Примерный вид конфигурации может быть следующим:

```yaml
global:
  scrape_interval:15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
  - job_name: 'spring-actuator'
    metrics_path: '/actuator/prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['192.168.0.9:8080']
```

Рис. 4. Конфигурационный файл prometheus.yml

После окончания всех настроек можно запустить с помощью Docker контейнеры с Prometheus и Grafana.

Теперь остается только настроить взаимодействие этих двух сервисов друг с другом. Это выполняется с помощью вкладки конфигурации по локальному адресу только что поднятого образа Grafana.

Дополнительным плюсом является то, что Grafana имеет большую библиотеку различных шаблонов dashboard, что позволяет не тратить время на создание и при этом дает возможность получить много необходимой информации для мониторинга. Добавление собственных метрик происходит с помощью интуитивно понятного интерфейса сервиса, важно лишь верно указать название созданной метрики.

На рис. 5 представлены некоторые метрики приложения, в том числе созданные ранее пользовательские (счетчик и датчик).

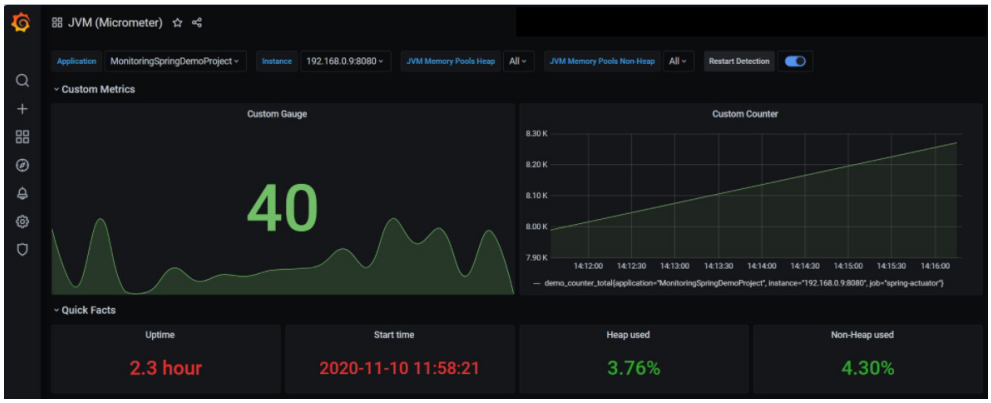


Рис. 5. Dashboard с пользовательскими метриками

Заключение

Подводя итог можно сказать, что мониторинг очень важен для современных приложений, так как по своей природе они сильно распределены и имеют различные зависимости: базы данных, службы, кеширование и многое другое. Собранные и визуально структурированные метрики позволяют выбрать путь по улучшению и оптимизации продукта, а также отследить дальнейший прогресс с учетом изменений.

Данная работа помогает понять для чего используется мониторинг, какие проблемы и задачи ставят перед собой разработчики, а также рассматривается одно из возможных решений интеграции мониторинга в существующее приложение с помощью Prometheus и Grafana.

Литература

1. Мониторинг Spring Boot с помощью Prometheus и Grafana (https://ordina-jworks.github.io/monitoring/2020/11/16/monitoring-spring-prometheus-grafana.html)

2. Документация Prometheus (https://prometheus.io/)

3. Документация Grafana (https://grafana.com/)

4. Grafana как еще один инструмент для технического мониторинга создаваемых нами программных продуктов (https://habr.com/ru/company/southbridge/blog/431122/)

Безрукова Ольга Алексеевна – студент 4-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: katlinc@mail.ru

Светлана Юрьевна Болотова (научный руководитель) – канд. физ.-мат. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: Bolotova.svetlana@gmail.com

УДК 519.85

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ

Н. С. Бурмистров, Е. М. Аристова

Воронежский государственный университет

Введение

Чрезвычайно широкий и крайне важный с практической точки зрения класс задач выбора составляют многокритериальные (многоцелевые) задачи, в которых качество принимаемого решения осуществляется по нескольким критериям.

В статье рассматриваются различные подходы для решения такого класса задач, и проводится сравнительный анализ эффективности этих подходов.

1. Постановка задачи многокритериальной оптимизации

Под задачей многокритериальной (многоцелевой векторной) оптимизации понимается задача вида:

$$F(X) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

с условиями:

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, j = 1, \ldots, k \tag{2}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, j = k+1, \ldots, m, \tag{3}$$

$$x_j \ge 0, j = 1, \ldots, l, l \le n, \tag{4}$$

где a_i, b_i, c_j – заданные константы.

Функция (1) называется *целевой функцией* или функцией цели задачи, а условия (2)–(4) называются *ограничениями* этой задачи.

Вектор $x = (x_1, \ldots, x_n)$, который удовлетворяет условиям (2)–(4), называется *допустимым решением*. Некоторое решение $x^* = (x_1^*, \ldots, x_n^*)$, при котором функция задачи (1) достигает своего максимального (минимального) значения, называется *оптимальным*.

Как показывает практика, критерии f_k оказываются противоречивыми, а это приводит к пустому пересечению множеств решений и к отсутствию идеального решения задачи. Поэтому формальная запись задачи многокритериальной максимизации не является основанием для решения, и считается, что задача многокритериальной оптимизации реализует априорные рациональные принципы оптимальности, которые полностью подходят под описание множества Ω положениями $f_k(x), k = 1, \ldots, M$ и предпочтительными направлениями изменения оценок f_k.

При решении задач многокритериальной оптимизации, в первую очередь, нужно найти некоторую область компромиссов, т. к. поиск решения и само решение будет находиться в этой области.

Область компромиссов – это некоторое подмножество из найденного множества оптимальных решений X, которое обладает таким свойством, что все принадлежащие ему решения не могут быть одновременно улучшены по всем локальным критериям.

Для того чтобы определить улучшаемость решения используется принцип Парето-оптимальности.

Определение 1. Точка x^0 называется *оптимальной по Парето* в задаче многокритериальной максимизации, если не существует другой точки $x' \in \Omega$, для которой $f_k(x') \geq f_k(x^0)$, $\forall k = 1,...,M$ и существует такой индекс k_0, что $f_k(x^0) > f_{k^0}(x^0)$.

Замечание. Заметим, что во множестве решений всех частных задач существует некоторое непустое подмножество точек, являющихся оптимальными по Парето, что, в свою очередь, обеспечивает существование решения задачи выбора с принципом эффективности по Парето. Более того, если решение каждой частной задачи x^k единственно, то $\{x^1,...,x^M\} \subset P_r$.

При решении задач многокритериальной оптимизации начальным шагом для решения следует считать выделение области оптимальных по Парето компромиссов (решений).

Каждый из векторов, принадлежащих множеству допустимых решений, который не является оптимальным по Парето, доминируется другим оптимальным вектором.

Решение является *оптимальным* тогда и только тогда, когда оно независимо от выбранного принципа оптимальности принадлежит области компромиссов. В противном случае, оно может быть улучшено и, соответственно, не будет являться оптимальным.

В связи с вышесказанным, область компромиссов – это область более узкая, чем область возможных решений, эта область содержит по сути своей потенциально оптимальные компромиссы. Поэтому следует искать решение задачи именно в области компромиссов, так как это значительно упрощает задачу из-за того, что область уже, чем вся область возможных решений.

Существуют также и дополнительные принципы оптимальности (принцип оптимальности по Слейтеру, принцип равновесия Нэша, принцип гарантированного результата, принцип оптимальности по Байесу и др.) [10].

В общем виде алгоритм решения задачи многокритериальной оптимизации можно схематично представить на рисунке [1]:

Рис. 1. Алгоритм решения задачи многокритериальной оптимизации

Пусть задача линейной многокритериальной оптимизации записана в виде:

$$\begin{cases} F(x) = (f_1(x),\ldots,f_n(x)) \to \max, \\ x \in X, \end{cases}$$

где $x \in X$, $X \in \mathbb{R}^n$ – это множество, состоящее из допустимых решений какой-то задачи многокритериальной оптимизации), а $F(x) = (f_1(x),\ldots,f_n(x))$, $f_i : R^n \to R$, $i = 1,\ldots,n$ – критерии.

Для решения задач многокритериальной оптимизации существует множество способов и методов, среди которых стоит отметить: принцип эффективности по Парето [1, 4, 7, 8], принцип приближения по всем локальным критериям к идеальному решению [4, 10], арбитражные решения [2, 8], целевое программирование [8], метод весовых коэффициентов [4, 5], метод приоритетов [8], метод последовательных уступок [9] и др.

2. Прикладная задача для исследования

Пусть задача имеет вид:

$$\begin{cases} f_1(x) = 4x_1 + 3x_2 \to \max \\ f_2(x) = 2x_1 + 4x_2 \to \max \\ f_3(x) = 3x_1 + 9x_2 \to \max \\ f_4(x) = 8x_1 + 2x_2 \to \max \\ f_5(x) = 3x_1 - 2x_2 \to \max \end{cases}, \tag{5}$$

при условиях

$$\begin{cases} 2x_1 + 3x_2 \le 18 \\ -x_1 + 3x_2 \le 9 \\ 2x_1 - x_2 \le 10 \\ x_1, x_2 \ge 0. \end{cases} \tag{6}$$

Для решения многокритериальных задач оптимизации воспользуемся существующими методами, результат представим в виде таблицы (табл. 1).

Таблица 1

Метод	**Полученная в ходе решения точка**	**Решение**
Метод ограничений	(5.87, 2.82)	$f(x^*) = (31.94, 23.02, 42.99, 52.6, 11.9).$
Арбитражная схема Нэша	(3, 4)	$f(x^*) = (24, 22, 45, 32, 17).$
Метод весовых коэффициентов	(6, 2)	$f(x^*) = (30, 20, 36, 52, 14).$
Метод последовательных уступок	(5.947, 1.894)	$f(x^*) = (29.47, 19.47, 34.887, 51.364, 14.053).$
Метод приближения по всем локальным критериям к идеальному решению	(5.686, 2.208)	$f(x^*) = (29.368, 20.204, 36.93, 49.904, 12.64).$

Здесь $f_i^*(x_i^*) = (f_i^*(x_1^*), \ldots, f_i^*(x_n^*))$, $i = 1 \ldots k$, где $x_i^* = (x_1^*, \ldots, x_i^*)$, $i = 1 \ldots k$ – вектор идеальных решений.

Автором был проведён сравнительный анализ эффективности указанных методов и получен следующий результат (табл. 2.).

Таблица 2

Метод	Достоинства	Недостатки
Метод ограничений	Значительное упрощение задачи из-за перехода от многокритериальной задачи (нескольких целевых функций) к однокритериальной (одной целевой функции).	Увеличение области допустимых решений может привести к неточному результату. Не подходит для решения задач с большим количеством целевых функций.
Арбитражная схема Нэша	Возможность повышения значений менее важных критериев.	Сложность (большая размерность) получаемой целевой функции.
Метод весовых коэффициентов	Простота реализации. Возможность не только расставить правильные приоритеты для критериев, но и учесть их в необходимых пропорциях.	Субъективность выбора весовых коэффициентов.
Метод последовательных уступок	Возможность повышения значений менее важных уступок. Возможность учета всех целевых функций с разной значимостью.	Сложность назначения и согласования значений уступок для каждого критерия. Субъективность выбора значений уступок.
Метод приближения по всем локальным критериям к идеальному решению	Какое-то решение будет найдено в любом случае.	Трудоёмкость вычислений при решении вручную.

Заключение

Исследования эффективности различных методов показывают, что не каждый метод применим к любой задаче многокритериальной оптимизации, и следует проводить анализ задачи для выбора наиболее подходящего метода, который бы удовлетворял всем условиям задачи.

В статье рассмотрены некоторые методы решения задач многокритериальной оптимизации (принцип эффективности по Парето, принцип приближения по всем локальным критериям к идеальному решению, арбитражные решения, целевое программирование, метод весовых коэффициентов, метод приоритетов, метод ограничений, метод последовательных уступок), а также описаны их достоинства и недостатки.

Это позволит грамотно определять, какой метод следует применять к решению конкретной задачи, что обеспечит более высокую эффективность этого решения.

Литература

1. *Подиновский В. В.* Парето-оптимальные решения многокритериальных задач / В. В. Подиновский, В. Д. Ногин. – 2-е изд., испр. и доп. – Москва : ФИЗМАТЛИТ, 2007. – 256 с.

2. *Кузютин Д. В.* Арбитражные решения в задачах выбора: методические указания. – Санкт-Петербург: Изд-во СПбГУ, 1995. – 92 с.

3. *Хазанова Л. Э.* Математическое моделирование в экономике: учебное пособие. –Москва: БЕК, 1998. – 141 с.

4. *Мелькумова Е. М.* О некоторых подходах к решению многокритериальных задач / Е. М. Мелькумова (Аристова) // Вестник Воронежского государственного технического университета. – Воронеж: ВГТУ, 2011. – том 7, №7. – С. 122–127.

5. *Зайченко Ю. П.* Исследование операций: учебное пособие для студентов вузов. – 2-е изд., перераб. и доп. – Киев : Вища школа, 1979. – 392 с.

6. *Домашева Д. В.* Методы решения задач многокритериальной оптимизации. – Оренбург: ГОУ ОГУ, 2008. – 49 с.

7. *Ануфриенко С. Е.* Метод Парето решения многокритериальных задач. – Ярославль: Ярославский гос. ун-т, 2004. – 24 с.

8. Задачи многокритериального линейного программирования. – Режим доступа: http://docplayer.ru/25848621-Nauchit-reshat-zadachu-mnogokriterialnogo-lineynogo-programmirovaniya.html (дата обращения: 22.04.2021).

9. Задачи многокритериальной оптимизации и методы их решения. – Режим доступа: https://studfile.net/preview/1676123 (дата обращения: 22.04.2021).

10. О некоторых подходах к решению многокритериальных задач. – Режим доступа: https://cyberleninka.ru/article/n/o-nekotoryh-podhodah-k-resheniyu-mnogokriterialnyh-zadach/viewer (дата обращения: 24.04.2021).

Бурмистров Никита Сергеевич – студент 1 курса магистратуры направления «Прикладная математика и информатика» Воронежского государственного университета. E-mail: onlyburik@gmail.com

Аристова Екатерина Михайловна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: pmim@yandex.ru

УДК ***

РОЛЬ ЛЕКСИЧЕСКОГО И СИНТАКСИЧЕСКОГО АНАЛИЗА В МАРШРУТЕ РАЗРАБОТКИ ПРОЕКТА ПЛИС

А. В. Варганов, Н. М. Малышев

Компания «ЭРЕМЕКС»

Введение

На сегодняшний день разработка проектов для конфигурации ПЛИС (программируемых логических интегральных схем) является достаточно востребованной тематикой. За счёт скорости работы, возможности многократной реконфигурации и простоты разработки ПЛИС находят применение в различных сферах – в телекоммуникации, космонавтике, в устройствах военного и гражданского применения [1, 2].

Важнейшее значение при разработке интегральных схем занимает процесс высокоуровневого описания схемы на языках разработки аппаратуры (HDL) и логическая верификация проекта. Кроме того, сложность верификации проекта увеличивается из года в год. Согласно исследованиям, проведённым компанией Mentor Graphics, на сегодняшний день на верификацию проектов уходит в среднем 50 % от времени, которое тратится на создание проекта [3]. Для высокоуровневого описания и верификации проектов могут быть использованы средства, предоставляемые САПР Delta Design Simtera, разрабатываемой компанией «ЭРЕМЕКС».

Наиболее популярными языками описания аппаратуры на сегодняшний день являются языки Verilog/SystemVerilog и VHDL. Заметим, что большую популярность приобретает язык SystemVerilog; это объясняется простотой его освоения в сравнении с VHDL, а также схожестью его синтаксиса с синтаксисом языка программирования C++. Благодаря этому приобретает актуальность вопрос поддержки языка SystemVerilog в программных средствах для разработки и верификации проектов ПЛИС.

Цель статьи заключается в исследовании места лексического и синтаксического анализа в процессе разработки ПЛИС, а также в изучении особенностей организации процесса лексического и синтаксического анализа проектов на языках описания аппаратуры Verilog и SystemVerilog с точки зрения разработчика САПР.

1. Маршрут разработки проекта ПЛИС

На рис. 1 показан маршрут разработки проекта программируемой интегральной схемы, который представляет собой итерационный процесс. При получении неудовлетворительных результатов на одном из этапов разработки предполагается возврат к этапу разработки высокоуровневого описания для доработки и получения необходимых характеристик.

После *описания* проекта на языке описания аппаратуры осуществляется его *логическая верификация*, которая представляет собой проверку правильности работы разработанного устройства и его соответствия требованиям, указанным в техническом задании.

Далее в процессе логического *синтеза* HDL-описание преобразуется в булевы функции. Современные синтезаторы строят принципиальные схемы реализаций этих функций, и разработчику необходимо тщательно анализировать эти схемы. Это необходимо для того, чтобы убедиться, что реализована желаемая логика.

После получения удовлетворительных результатов синтеза производится *отображение* функций на логические элементы конкретной схемы и *проектирование топологии*. Если вы-

ясняется, что проект слишком велик, чтобы уместиться в выбранной ПЛИС, его приходится пересматривать.

Решив топологические задачи, проводится *временная верификация* – ограничения во временной области (например, заданная тактовая частота) сравниваются с реальными задержками, получаемыми в схеме. Может оказаться, что схема работает слишком медленно, из-за чего её необходимо модернизировать. После исправления всех ошибок генерируется конфигурационный файл, отражающий полную структуру разработанного проекта [4].

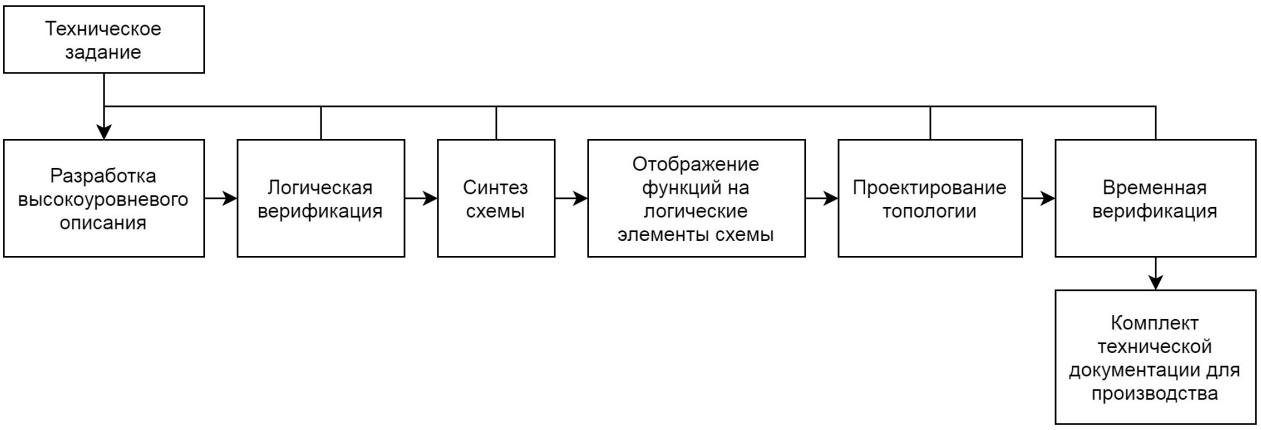


Рис. 1. Маршрут разработки проекта ПЛИС

2. Логическая верификация. Лексический и синтаксический анализ

На рис. 2 представлен процесс логической верификации проекта ПЛИС с точки зрения разработчика САПР. *Логическая верификация* делится на два больших этапа – компиляцию проекта и его моделирование. *Компиляция* проекта, в свою очередь, включает в себя шесть последовательных процессов [5].

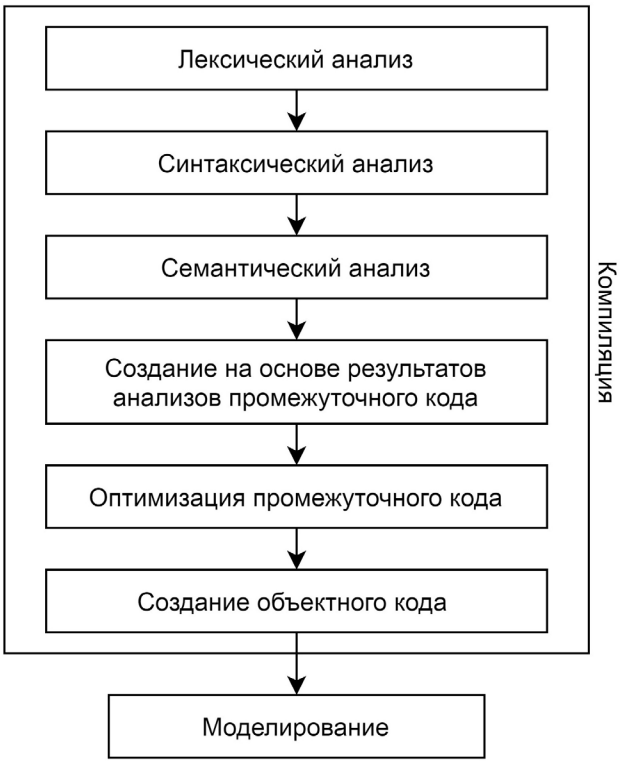


Рис. 2. Процесс логической верификации проекта ПЛИС

Лексический анализ представляет собой разбиение текста программы на слова (лексемы) и формирование упорядоченного списка лексем. В ходе *синтаксического анализа* сформированный список сопоставляется с грамматикой языка, на котором была написана программа, в результате чего строится абстрактное синтаксическое дерево (AST, abstract syntax tree), отражающее грамматическую структуру проекта. Полученное дерево подвергается *семантическому анализу* – процессу проверки исходного текста программы на семантическую согласованность со стандартом языка. В частности, на этапе семантического анализа компилятор проверяет, имеет ли каждый оператор операнды соответствующего типа.

На данных трёх этапах компилятором выявляются ошибки в тексте проекта. Ошибки делятся на три типа (в зависимости от этапа, на котором они выявлены) – *лексические*, *синтаксические* и *семантические*.

Последующие три этапа компиляции часто объединяют в один этап, который называется кодогенерацией. Во время *кодогенерации* генерируется абстрактное промежуточное представление исходного кода, которое подвергается оптимизации и в итоге преобразуется в целевой код. В роли целевого кода может выступать как машинный код, так и код на любом объектно-ориентированном языке программирования. В системе Delta Design Simtera программы на языках описания аппаратуры VHDL и SystemVerilog при компиляции преобразуются в код на языке программирования C#.

В том случае, если компиляция проекта прошла успешно, осуществляется *моделирование* скомпилированного проекта. На входы разработанной схемы подаются тестовые воздействия, и выходные сигналы сравниваются с ожидаемыми. В случае различия между реальными и ожидаемыми результатами работы схемы логическая верификация считается неудачной, после чего следует вернуться на этап описания схемы для её отладки.

3. Проблемы, возникающие при реализации лексического и синтаксического анализатора проектов на языке SystemVerilog, и их решение

Главной особенностью этапа логической верификации является жёсткое следование языковым стандартам; в частности, стандартам языка SystemVerilog. Однако при строгой поддержке данных стандартов во время реализации этапов лексического и синтаксического анализа возникают несколько проблем, а именно:

- Использование компиляционных директив в коде проекта;
- Возникновение взаимной левой рекурсии в грамматике синтаксического анализатора (парсера).

3.1. Использование компиляционных директив в HDL-коде

Одной из особенностей языка описания аппаратуры SystemVerilog является использование в коде *компиляционных директив*. Такие директивы начинаются с символа «`» [6].

Тем не менее, использование директив в коде HDL-проекта увеличивает количество времени, требуемого процессору для лексического и синтаксического разбора. Это обусловлено тем, что для обработки компиляционных директив требуется введение дополнительного, предварительного уровня лексического разбора. Этот уровень необходим для того, чтобы выделить из кода проекта текст директивы и провести его подробный анализ. Кроме того, необходимо подготовить входные данные к следующему уровню разбора – например, если в проекте используется директива define и в коде существуют ссылки на данную директиву, необходимо на предварительном уровне лексического разбора заменить эти ссылки на значение, указанное в тексте директивы. После этого на основном уровне лексического разбора осуществляется «классическое» разбиение текста программы на лексемы.

В табл. 1 представлен простой пример проекта на языке SystemVerilog с использованием директив define и без их использования. С помощью системы тестирования работы САПР Delta Design Simtera был произведён стократный разбор данных проектов с замером процессорного времени, потребовавшегося на разбор. На основе полученных данных было вычислено среднее время сборки проектов. Как видно на рис. 3 и 4, проект, в котором использовались компиляционные директивы, разбирается дольше аналогичного проекта без директив. Важно также понимать, что время сборки проекта зависит не только от количества используемых директив, но и от объёма кода.

Таблица 1

Пример проекта на языке описания аппаратуры SystemVerilog

Без использования компиляционных директив

```
module Counter56 (POR, CLK, VoltageOK, ChargeDone);
input POR;
input CLK;
input VoltageOK;
output ChargeDone;

reg [1:0] State, nextState;
wire [8:0] nextMinuteCounter;

always @(posedge CLK or negedge POR)
    if (!POR)
        State = 2'b00;
    else
        State = nextState;

always @(VoltageOK)
    casez (State)
        2'b00: begin
            nextState = (VoltageOK) ? 2'b01 : 2'b00;
        end
        2'b01: begin
            nextState = (VoltageOK) ? 2'b01 : 2'b00;
        end
        2'b10: begin
            nextState = `DONE;
        end
        default: begin
            nextState = 2'bxx;
        end
    endcase
endmodule
```

С использованием директив define

```
`define IDLE 2'b00
`define CNT 2'b01
`define DONE 2'b10

module Counter56 (POR, CLK, VoltageOK, ChargeDone);
input POR;
input CLK;
input VoltageOK;
output ChargeDone;

reg [1:0] State, nextState;
wire [8:0] nextMinuteCounter;

always @(posedge CLK or negedge POR)
    if (!POR)
        State = 2'b00;
    else
        State = nextState;

always @(VoltageOK)
    casez (State)
        `IDLE: begin
            nextState = (VoltageOK) ? `CNT : `IDLE;
        end
        `CNT: begin
            nextState = (VoltageOK) ? `CNT : `IDLE;
        end
        `DONE: begin
            nextState = `DONE;
        end
        default: begin
            nextState = 2'bxx;
        end
    endcase
endmodule
```

Рис. 3. Время разбора проекта в системе Delta Design Simtera

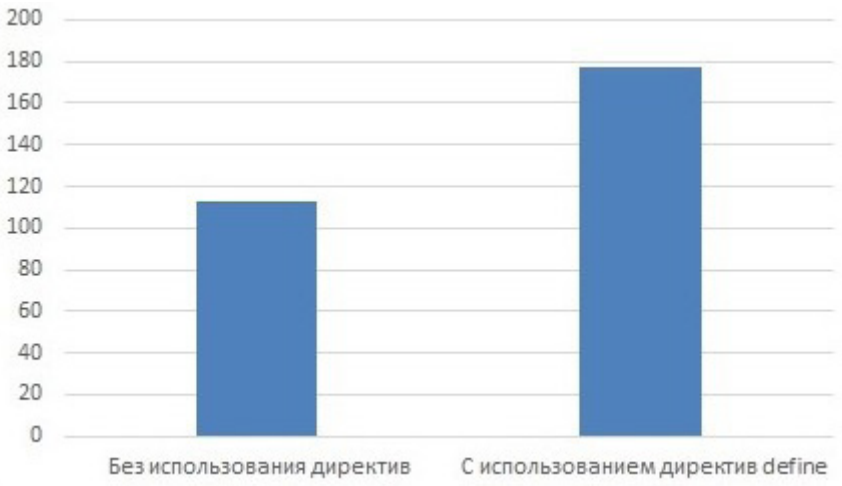


Рис. 4. Среднее время разбора проекта в миллисекундах

3.2. Устранение взаимной левой рекурсии

При реализации синтаксического анализатора (парсера) на этапе описания синтаксиса конструкций языка может возникнуть ситуация, когда несколько правил ссылаются друг на друга таким образом, что при работе парсера возникает ошибка, хотя с точки зрения грамматики всё описано строго согласно стандарту. Такая ошибка возникает из-за наличия *взаимной левой рекурсии*, которую не поддерживают средства разработки анализаторов. Данная ошибка может быть устранена путём сокращения правил, однако это может значительно повлиять на качество синтаксического анализа. В лучшем случае может быть построено неправильное абстрактное синтаксическое дерево; в худшем же случае парсер будет находить ошибку в тексте анализируемого проекта там, где её на самом деле нет.

Однако данная проблема может быть решена путём перестановки или объединения правил и подправил, при этом все эти изменения в грамматике парсера должны осуществляться таким образом, чтобы конечная компоновка правил соответствовала принятому стандарту. В качестве примера рассмотрим конструкции expression и inside_expression, синтаксис которых приведён на рис. 5 [6].

```
expression ::=
        primary
      | unary_operator { attribute_instance } primary
      | inc_or_dec_expression
      | ( operator_assignment )
      | expression binary_operator { attribute_instance } expression
      | conditional_expression
      | inside_expression
      | tagged_union_expression

inside_expression ::= expression inside { open_range_list }
```

Рис. 5. Синтаксис конструкций expression и inside_expression языка описания аппаратуры SystemVerilog

При дословном переносе данных конструкций в грамматику парсера возникает взаимная левая рекурсия, так как правило expression в одном из подправил полностью ссылается на правило inside_expression, которое в качестве первого элемента имеет ссылку на expression. В итоге получается зацикливание, которое следует разорвать путём переноса синтаксиса правила inside_expression в одно из подправил expression. После выполнения данных действий синтаксис правила expression будет иметь вид, представленный на рис. 6.

```
expression ::=
        primary
      | unary_operator { attribute_instance } primary
      | inc_or_dec_expression
      | ( operator_assignment )
      | expression binary_operator { attribute_instance } expression
      | conditional_expression
      | expression inside { open_range_list }
      | tagged_union_expression
```

Рис. 6. Синтаксис конструкции expression языка описания аппаратуры SystemVerilog после устранения взаимной левой рекурсии

Отметим, что это далеко не единственный случай, когда в конструкциях парсера возникала взаимная левая рекурсия. Однако было доказано, что описанный выше метод эффективно работает во всех случаях, когда при написании грамматики синтаксического анализатора возникала проблема взаимной левой рекурсии.

4. Зависимость времени разбора проекта от его размера

Отдельной задачей при разработке САПР является тестирование разработанного программного обеспечения. Тестирование работы алгоритмов разбора проектов и функционала программы осуществляется с помощью специально разработанной системы тестирования. Данная система тестирования в автоматическом режиме запускает сборку и моделирование тестовых проектов на языках VHDL и Verilog/SystemVerilog, в которых описаны различные модули, функции и прочие тестируемые элементы.

С помощью системы тестирования САПР Delta Design Simtera были получены данные, отражающие зависимость времени разбора проекта от количества лексем в его тексте. Данные зависимости показаны на рис. 7 и 8. Каждая точка на этих рисунках – это HDL-файл, описывающий работу цифрового устройства и/или его части.

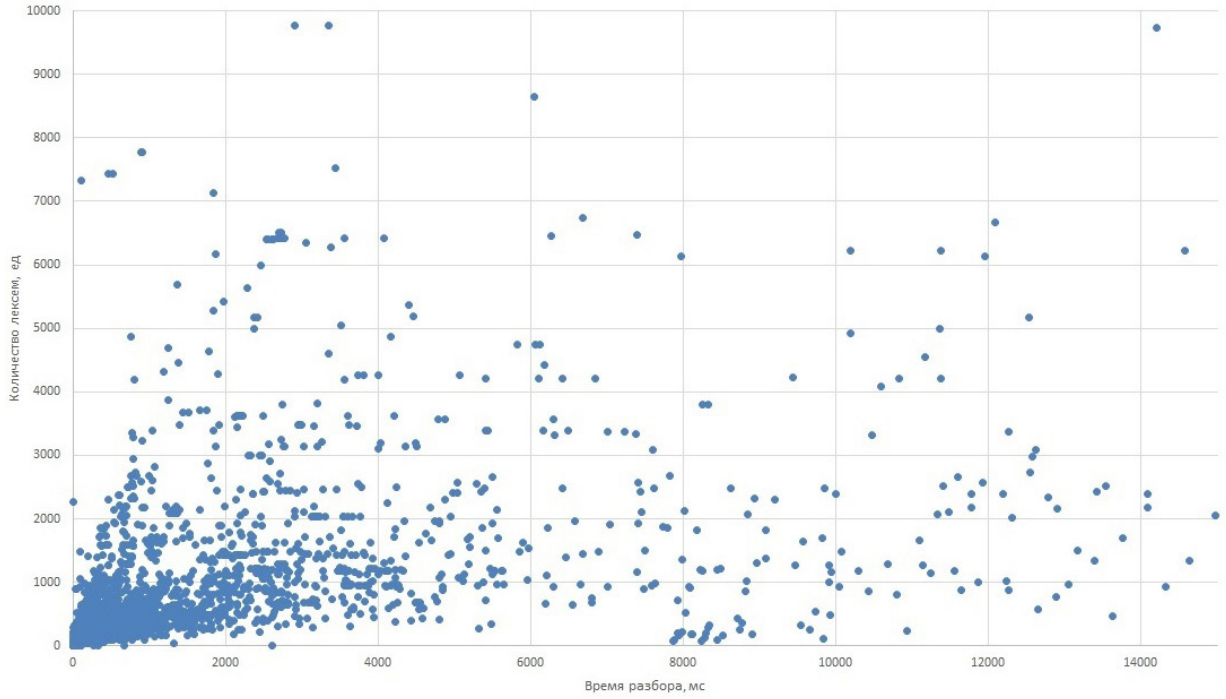


Рис. 7. Распределение тестов при разборе входных данных (общий вид)

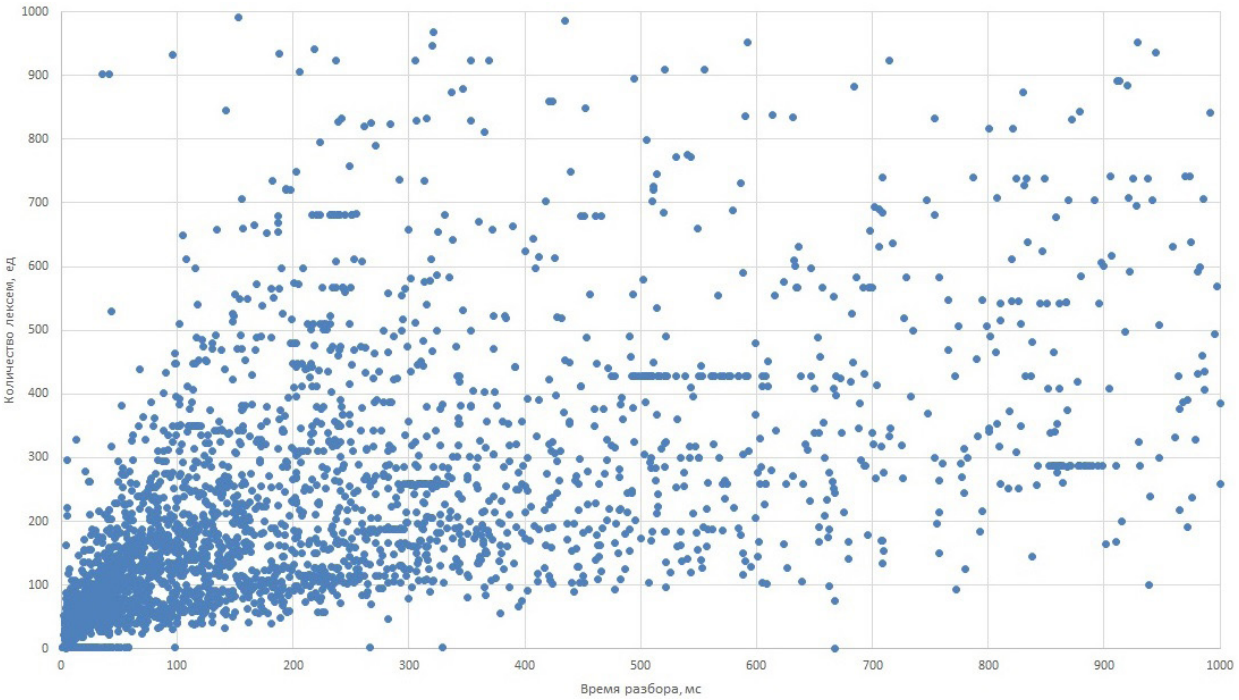


Рис. 8. Распределение тестов при разборе входных данных (в масштабе)

Детального рассмотрения требуют точки, лежащие на оси абсцисс. Более подробный анализ показывает, что данные точки представляют собой файлы, в которых содержатся только компиляционные директивы. Такие файлы, как правило, включаются в проекты устройств с помощью директивы include. Как уже было отмечено ранее, разбор таких файлов требует большего количества процессорного времени, чем файлы без использования компиляционных директив.

Заключение

В настоящей статье была подробно рассмотрена роль логической верификации в маршруте разработки проекта ПЛИС, а также было определено место лексического и синтаксического анализа в процессе логической верификации.

Были рассмотрены проблемы, возникающие при разработке лексического и синтаксического анализатора проектов на языке описания аппаратуры SystemVerilog. В частности, была наглядно продемонстрирована зависимость времени разбора проекта от использования в коде проекта компиляционных директив.

Литература

1. *Малышев, Н. М.* Средства функциональной верификации компании Eremex / Н. М. Малышев // Современная электроника. – 2018. – № 7. – С. 36–37.

2. *Малышев, Н. М.* Особенность разработки САПР для проектирования и верификации конфигурации ПЛИС / Н. М. Малышев, С. В. Рыбкин // Международный форум «Микроэлектроника-2019». Сборник тезисов. – 2019. – С. 278–281.

3. Part 3: The 2018 Wilson Research Group Functional Verification Study. – URL: https://blogs.sw.siemens.com/verificationhorizons/2018/12/04/part-3-the-2018-wilson-research-group-functional-verification-study. – (Дата обращения: 25.03.2021).

4. *Харрис, Дэвид М.* Цифровая схемотехника и архитектура компьютера / Дэвид М. Харрис, Сара Л. Харрис. – 2-е изд. – Morgan Kaufman, 2013. – 1662 с.: ил.

5. *Ахо, Альфред В.* Компиляторы: принципы, технологии и инструментарий / Альфред В. Ахо, Моника С. Лам, Сети, Рави Сети, Джеффри Д. Ульман. – 2-е изд. : Пер с англ. – М. : ООО «И.Д. Вильямс», 2018. – 1184 с.: ил. – Парал. тит. англ.

6. IEEE Std 1800–2005. IEEE Standard for SystemVerilog – Unified Hardware Design, Specification and Verification Language. – New York, IEEE, 2005. – 648 p.

Варганов Артем Вадимович – ведущий разработчик направления компании «ЭРЕМЕКС», магистрант 2-го года обучения кафедры 304 «Вычислительные машины, системы и сети» Московского авиационного института. E-mail: varganov@eremex.ru

Малышев Никита Максимович – ведущий разработчик компании «ЭРЕМЕКС». E-mail: malyshev.n@eremex.ru

УДК 004.942

ТЕСТИРОВАНИЕ АЛГОРИТМА РАСПОЗНАВАНИЯ МУЗЫКИ НА ОСНОВЕ ОТПЕЧАТКОВ АУДИО

А. В. Васильев, М. К. Чернышов

Воронежский государственный университет

Введение

Иногда возникает потребность узнать название песни, которая играет в каком-либо общественном месте, по радио, телевизору, или в Вашей музыкальной коллекции есть песни, автор и название которых не известны. Для того чтобы узнать название песни нужна программа для распознавания музыки. Распознавание музыки, или песни – это процесс идентификации музыкальной композиции по некоторой небольшой части (образцу) её цифрового аудиосигнала.

В данной статье приведены результаты тестирования клиент-серверного приложения, которое реализует алгоритм распознавания музыки на основе отпечатков аудио. Идея алгоритма была взята из статьи [1] Кристофа Калензага.

1. Алгоритм

Первым шагом к получению аудио отпечатков каждой композиции является **построение ее частотного спектра**. На рис. 1 представлен один из возможных алгоритмов для получения спектра аудиосигнала с точки зрения клиент-серверного приложения.

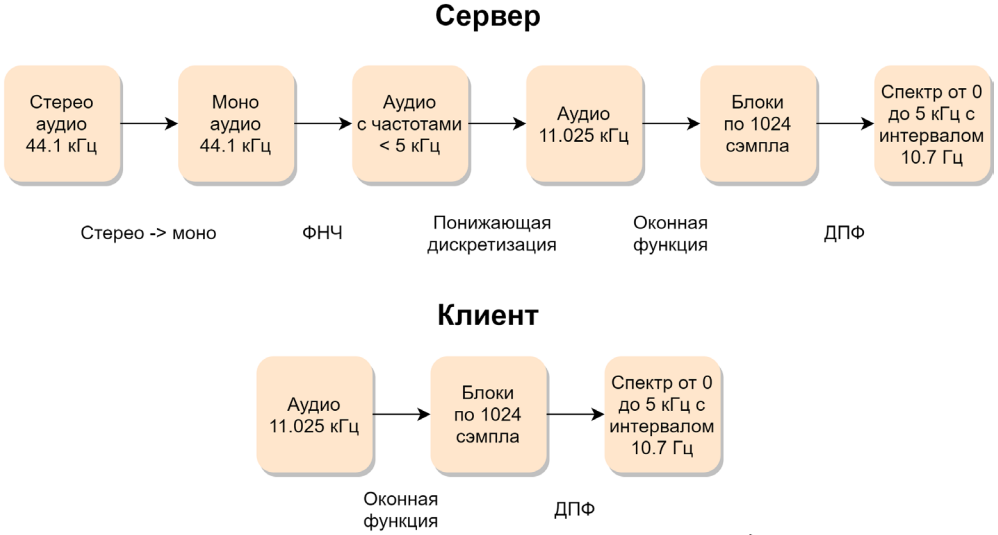


Рис. 1. Алгоритм получения частотного спектра аудио композиции на стороне клиента и на стороне сервера

Поскольку алгоритм распознавания музыки должен быть устойчив к шуму, на этапе **фильтрации спектра** необходимо учитывать только самые сильные (имеющие наибольшую амплитуду в спектре) частоты. Вместо использования n наиболее мощных частот в спектре, необходимо разделить весь доступный спектр на частотные интервалы (например, по октавам).

Будем называть частотно-временной точкой упорядоченную пару значений времени и частоты в спектре сигнала. Вместо того, чтобы в тестовом образце исследовать каждую такую точку по очереди, необходимо обрабатывать и искать в композициях сразу несколько из них.

Такую группу точек будем называть целевой зоной. Этап **создания отпечатков песен** заключается в генерации массива, элементами которого является опорная точка вместе с её целевой зоной (рис. 2).

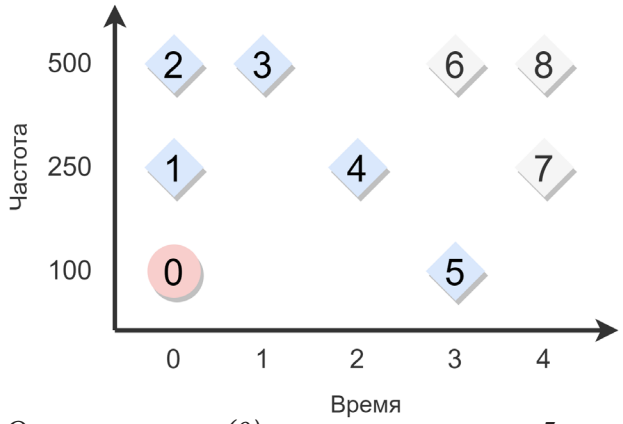


Рис. 2. Опорная точка (0) и ее целевая зона из 5 точек (1-5)

2. Программная реализация

Данный алгоритм реализован в виде клиент-серверного приложения, общая архитектура которого представлена на рис. 3.

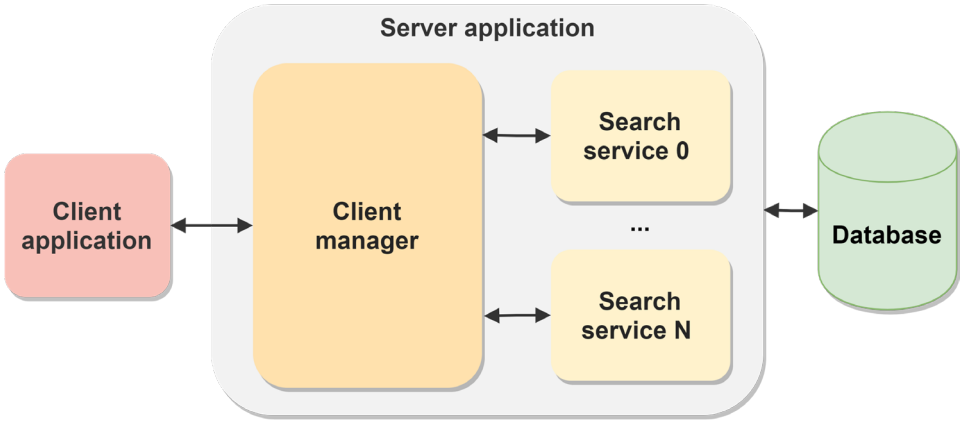


Рис. 3. Архитектура клиент-серверного приложения для распознавания музыки

В состав приложения входит:

1. **Клиентское приложение.** В данной статье в качестве него выступает специальная программа для тестирования, генерирующая необходимые данные на вход серверного приложения.
2. Серверная часть состоит из двух элементов:
   2.1. **Сервис обработки пользовательских запросов** (client manager) и индексирования песен.
   2.2. **Сервис поиска** (search service), имеется возможность распараллеливания задачи поиска с помощью запуска нескольких экземпляров сервиса.
3. **База данных** (database). Используется NoSQL база данных MongoDB.

3. Тестирование

Все тесты проводились на следующей конфигурации аппаратного обеспечения:

1. CPU: Intel Xeon X3470 (4 ГГц, 4 ядра, 8 потоков), теоретическая производительность – 9 ГФлопс.

2. RAM: 24 Gb DDR3 1600MHz.

3. OS: Windows 10 (build 2004) x64.

Подбор начальных параметров. Прежде всего нужно найти наиболее оптимальные значения констант и коэффициентов, необходимых для работы алгоритма распознавания. Каждому параметру задается список приблизительных значений, и тест повторяется для всех комбинаций. Общее их количество составило 648, на каждой итерации осуществлялся поиск по 20 записанным на микрофон сэмплам длиной 4 секунды (только 10 исходных композиций которых присутствовали в БД), а количество проиндексированных музыкальных композиций – 320. Наиболее оптимальные параметры были найдены на основе трех основных результирующих величин – правильность «угадывания» песни, время поиска и объем базы данных (табл. 1).

Таблица 1

Оптимальные значения параметров для алгоритма распознавания

Параметр	**Значение**
Размер целевой зоны	3
Погрешность целевой зоны (допустимая величина несовпадения целевых зон записи и музыкальной композиции)	1
Коэффициент отбрасывания частот в спектре (этап фильтрации, дополнительно умножается на среднее от амплитуд частот по всем диапазонам)	0.5
Размер окна БПФ	1024
Частотные интервалы	130–261 Гц 261–523 Гц 523–1046 Гц 1046–2093 Гц 2093–4186 Гц
Порог совпадающих целевых зон в записи и песне	0.5
Порог совпадения записи и песни (после согласования по времени)	6.5

Данные параметры будут использованы при следующих этапах тестирования.

Тестирование на базе данных большего размера (2300 песен). Использовался набор записанных сэмплов из предыдущего этапа. Были получены следующие результаты:

1. Средний объем 1 песни в БД (в виде отпечатков) приблизительно равен 4 Мб;

2. Среднее время поиска (на заданной аппаратной конфигурации, при 4 потоках поискового сервиса) – 2.3 секунды;

3. Правильность результата (рис. 4). 60 % – правильный ответ (песня угадана верно, либо песни нет в БД), 15 % – ничего не найдено, 25 % – неправильный ответ (найдена неправильная песня).

Для аналогичных сэмплов (но уже длиной 10 секунд), правильность результата увеличилась с 60 % до 71 %, а соотношение между «неверно» и «не найдено» сместилось в сторону второго пункта, что можно увидеть на рис. 5.

Устойчивость к шуму. Данное тестирование отражает устойчивость алгоритма с заданными параметрами к постороннему шуму в записанных сэмплах. В табл. 2 представлены результаты, сгруппированные по разным условиям записи. Для каждого случая было произведено по 20 записей сэмплов (длительностью 10 секунд) различных музыкальных композиций. В качестве фонового шума использовалась запись звука, созданная в общественном месте, которая суммировалась с полезным сигналом в разных пропорциях.

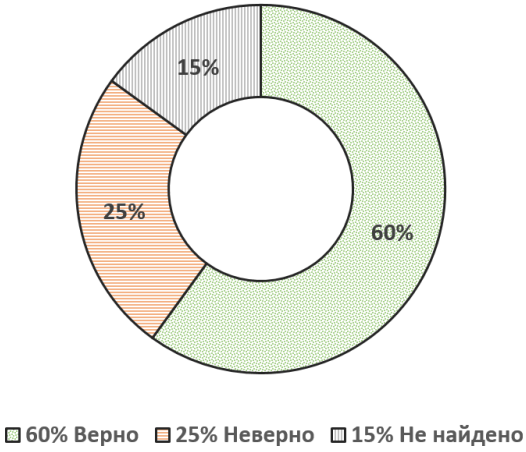


Рис. 4. Диаграмма результатов тестирования для сэмплов длиной 4 секунды

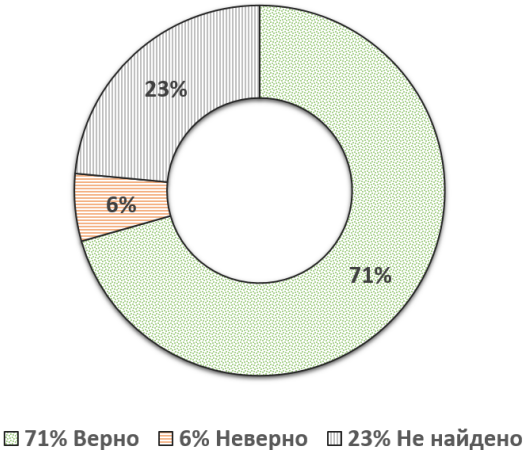


Рис. 5. Диаграмма результатов тестирования для сэмплов длиной 10 секунд

Таблица 2

Результаты тестирования на устойчивость к шуму

Источник звука и устройство записи	**Процент правильных ответов, %**
В одном помещении, без шума	71
В одном помещении, небольшой шум	71
В одном помещении, значительный шум	65
В одном помещении, шум громче полезного сигнала	35

Заключение

Из результатов тестирования можно сделать следующие выводы:

1. Алгоритм показал от 60 % до 70 % правильности результата в зависимости от длительности сэмпла (даже при наличии посторонних шумов);

2. При наличии шума с уровнем громкости выше уровня громкости источника сигнала алгоритм предсказуемо не обеспечивает достаточной степени правильности ответов, поскольку сильные частоты в спектре шума перекрывают полезный сигнал, однако всё ещё способен правильно распознать песню с вероятностью около 30–40 %;

3. При увеличении длины сэмпла правильность результата предсказуемо растет. Для найденного набора параметров оптимальным значением является интервал 8–10 секунд. При длине сэмпла около 4-х секунд велик процент ложного срабатывания (т.е. алгоритм выдает за верный ответ неправильную песню);

4. Алгоритм требователен к объему оперативной памяти, и при большом количестве песен в базе данных могут потребоваться значительные объемы серверных ресурсов. Однако задача поиска легко масштабируется, что облегчает задачу наращивания мощностей.

Литература

1. How does Shazam work. – Режим доступа: http://coding-geek.com/how-shazam-works. – (Дата обращения: 07.11.20).

2. *Кинтцель, Т.* Руководство программиста по работе со звуком / Кинтцель Т. –Москва : ДМК Пресс, 2000. – 432 с.

3. Лафоре, Р. Структура данных и алгоритмы в Java / Р. Лафоре. – Питер, 2018. – 704 с.

Васильев Александр Вадимович – студент 1-го курса магистратуры кафедры математического обеспечения ЭВМ воронежского государственного университета.
E-mail: ledkast@gmail.com

Чернышов Максим Карнельевич (научный руководитель) – канд. физ-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: mkch69@gmail.com

УДК 004.932.4

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ БЕСШОВНОГО НАЛОЖЕНИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ РЕШЕНИЯ УРАВНЕНИЯ ПУАССОНА

И. А. Веселов, О. Д. Горбенко

Воронежский государственный университет

Введение

На рис. 1 продемонстрированы обозначения:

$A \in R^2$ – замкнутое множество, является областью определения изображения;

B – замкнутое подмножество A с границей ∂B;

f – известная скалярная функция на $A \setminus \mathrm{int}(B)$, описывающая изображение, на которое накладываем другое изображение;

g – неизвестная скалярная функция на $\mathrm{int}(B)$;

h – известная функция на $\mathrm{int}(B)$, описывающее накладываемое изображения;

v – направляющее векторное поле.

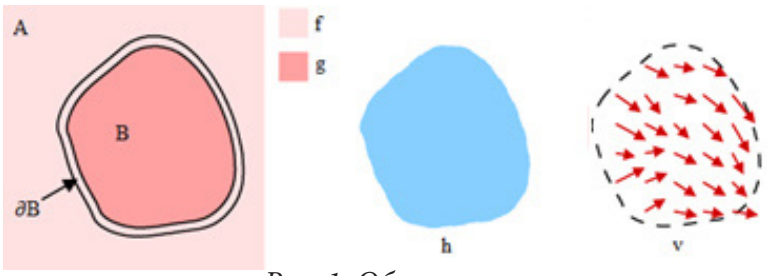


Рис. 1. Обозначения

Для бесшовного наложения изображений требуется решить уравнение Пуассона с граничными условиями Дирихле для каждого цветового канала изображения [1]:

$$\Delta g = \mathrm{div}\, v, \quad g\big|_{\partial B} = f\big|_{\partial B}, \tag{1}$$

где $\mathrm{div} = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}$ – оператор дивергенции.

В качестве направляющего поля v для бесшовного наложения можно взять градиентное поле функции h, которое показывает направление наибольшего возрастания значения цвета, тогда $\mathrm{div}\, v = \Delta h$ [1, 2].

Существуют задачи, когда такое бесшовное наложение не дает необходимый результат. Например, требуется наложить только часть накладываемого изображения или сохранить часть изображения, на которое производится наложение. Для таких задач требуются модификации данного алгоритма.

1. Модификации алгоритма для различных задач

1.1. Наложение шаблона интенсивности цвета

Часто встречается задача наложить шаблон интенсивности цвета, но не сам цвет. Для решения этой задачи необходимо предварительно сделать накладываемое изображение монохромным. (рис. 2в). На рис. 2д представлен результат наложения монохромного изображения. В отличие от рис. 2г на рис. 2д отсутствует более интенсивный красный цвет.

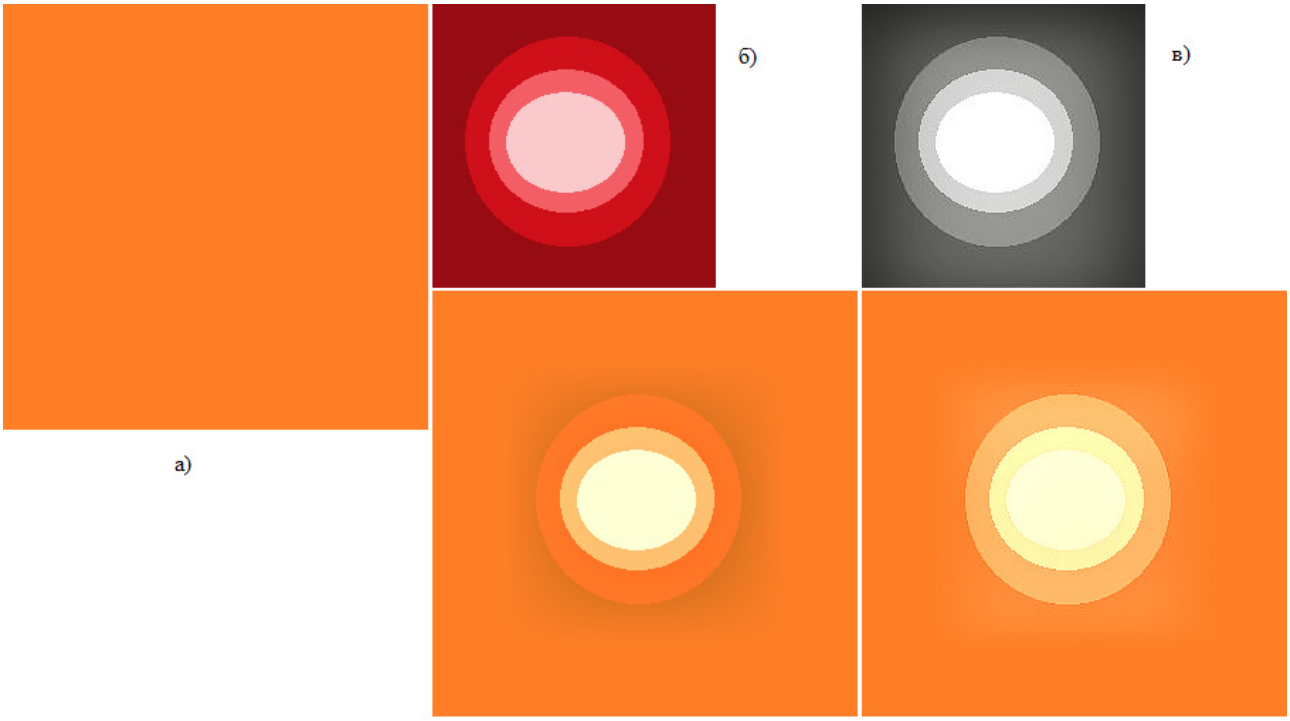


Рис. 2. Наложение шаблона интенсивности цвета а) исходное изображение, б) накладываемое изображение, в) монохромное накладываемое изображение, г) бесшовное наложение, д) бесшовное наложение с использованием монохромного

1.2 Смешивание градиентов

При бесшовном наложении результат g на $\mathrm{int}(B)$ не содержит шаблона интенсивности цвета, который присутствует на изображении, на которое производится наложение. Однако, может быть необходимо сохранить свойства обоих исходных изображений. Например, наложение объектов с дырами или частично прозрачных, когда необходимо сохранить текстуру фона.

Один из возможных подходов – определить направляющее поле v как линейную комбинацию градиентных полей функций h и f.

$$v(x) = a\nabla h(x) + b\nabla f(x), \quad x \in B, \quad a \in (0,1), \; b = 1 - a. \tag{2}$$

Но данный подход не дает достаточно качественный результат. На рис. 4 представлен результат бесшовного наложения с определением направляющего поля как линейной комбинации градиентов.

Рис. 3. Пример бесшовного наложения а) исходное изображение, б) накладываемое изображение, в) результат бесшовного наложения

Рис. 4. Бесшовное наложение с использованием линейной комбинации

Бесшовное наложение изображений, основанное на решении уравнения Пуассона, позволяет использовать нестандартные направляющие поля, что позволяет добиться лучшего эффекта. Для каждой точки из B будем брать наибольший по модулю градиент функций h и f:

$$v(x) = \begin{cases} \nabla h(x), |\nabla h(x)| \geq |\nabla f(x)| \\ \nabla f(x), |\nabla h(x)| < |\nabla f(x)| \end{cases}, \quad x \in B. \tag{3}$$

На рис. 5, 6 продемонстрированы результаты бесшовного наложения с использованием максимального градиента.

Рис. 5. Бесшовное наложение с использованием максимального градиента

Рис. 6. Бесшовное наложения с прозрачным объектом а) исходное изображение, б) накладываемое изображение, в) результат бесшовного наложения г) результат бесшовного наложения с использованием максимального градиента

Использование смешанного градиента также полезно при вставке объекта близко к другому объекту на изображении.

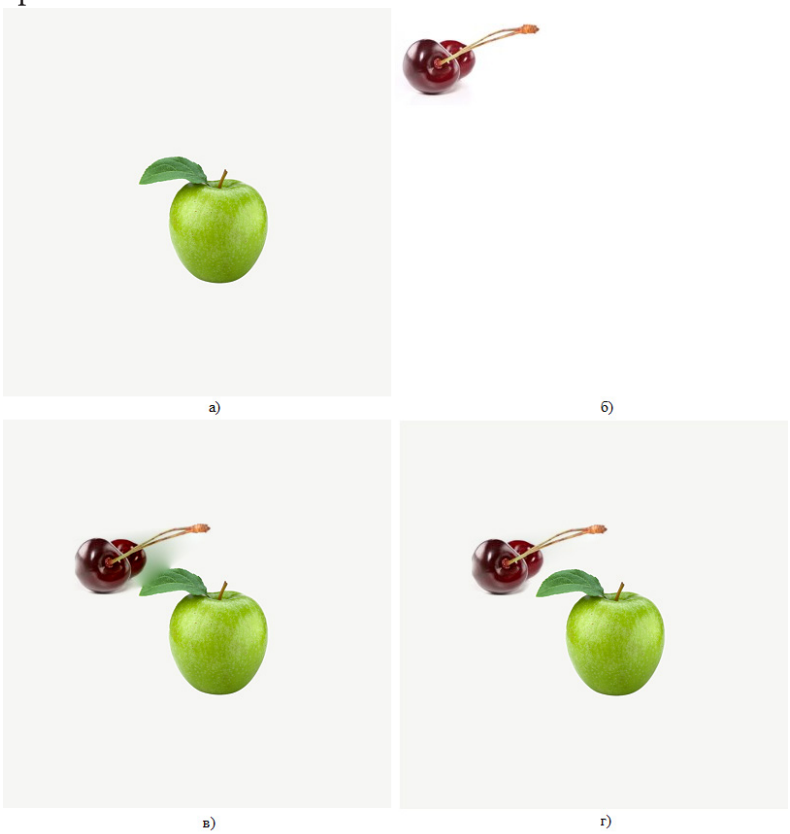


Рис. 7. Бесшовное наложения объектов близко друг к другу а) исходное изображение, б) накладываемое изображение, в) результат бесшовного наложения г) результат бесшовного наложения с использованием максимального градиента

Заключение

В статье продемонстрированы возможности решения различных задач бесшовного наложения изображений с использованием преобразования Пуассона и модификаций алгоритма. Данные модификации алгоритма позволяют накладывать только нужную часть изображения, либо сохранять необходимые элементы исходного изображения.

Литература

1. *Веселов, И. А.* Бесшовное наложение изображений с помощью уравнений Пуассона / И. А. Веселов // Межвузовская научная конференция молодых ученых и студентов «Математика, информационные технологии, приложения!. – 2020. – С. 40–42.

2. *Pérez, P.* Poisson Image Editing / P. Pérez, M. Gangnet, A. Blake // ACM Transactions on Graphics. – 2003. – V. 22, No. 3. – P. 313–318.

3. *Гонсалес, Р. С.* Цифровая обработка изображений / Р. С. Гонсалес, Р. Е. Вудс. – 3-е изд., исправл. и доп. – Москва : Изд-во Техносфера, 2012 – 1104 с.

4. *Загарян, Ю. А.* Компьютерная графика в практических приложениях / Ю. А. Загарян, Е. В. Загарян, – Таганрог : Изд-во ТТИ ЮФУ, 2009 – 255 с.

5. *Цисарж, В. В.* Математические методы компьютерной графики / В. В. Цисарж, Р. И. Марусик. – Киев : Изд-во Факт, 2004. – 464 с.

6. *Павлидс, Т.* Алгоритмы машинной графики и обработки изображений / Т. Павлидс – Изд-во Радио и связь, 1986. – 400 с.

7. *Прэтт, У.* Цифровая обработка изображений. Книга 1 / У. Прэтт – Москва : Изд-во Мир, 1982. – 311 с.

8. *Прэтт, У.* Цифровая обработка изображений. Книга 2 / У. Прэтт – Москва : Изд-во Мир, 1982. – 479 с.

9. Применение преобразования Пуассона для бесшовного наложения изображений: http://habr.com/ru/post/213403/ (дата обращения: 18.04.2021).

Веселов Илья Александрович – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: ves.ilya@mail.ru

Горбенко Олег Данилович (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: oleg_dan@mail.ru

УДК 004.932

ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ

О. С. Воронова

Воронежский государственный университет

Введение

Ежедневно каждый человек сталкивается с различными изображениями. Они стали неотъемлемой частью жизни. Все важные и памятные моменты запечатлеваются с помощью различных устройств (фотоаппарат, телефон и т. д.). В век цифровых технологий хранение, обработка и передача изображений стала значительно эффективнее и проще. Однако осталось огромное количество фотографий, которые хранятся на бумажном носителе. Со временем изображения выцветают, появляются потёртости, трещины и другие дефекты. Одной из актуальных проблем можно назвать восстановление оцифрованных фотографий. Под этим процессом понимается улучшение качества, удаление шумов и искажений, устранение механических повреждений. Главная сложность и особенность данной задачи заключается в том, что универсального и единого метода нет.

1. Анализ задачи

Один из ключевых моментов, который определяет необходимость изучения методов и способов решения данной задачи, заключается в широком распространении и применении изображений. На сегодняшний день известны методики обработки фотографических данных вручную и с использованием технологий [1]. Однако, данный вариант требует огромных трудозатрат как временных, так и ресурсных, в процессе можно повредить или окончательно испортить обрабатываемые фотографии.

Задачу цифровой обработки можно решить, применяя специализированные программные продукты, такие как GIMP, Adobe Photoshop, Paint.NET. В данной ситуации все действия приходится выполнять вручную, что усложняет процесс работы с изображением.

Возникает необходимость использования технологий, которые знают и умеют эффективно работать с повреждёнными цифровыми объектами. Наибольшую популярность в последнее время приобретают нейронные сети, которые доказали свою эффективность при выполнении задач различной сложности. Их применение при в области цифровой обработки изображений позволит значительно повысить точность результатов и уменьшить временные и трудовые затраты.

2. Нейронные сети

Решать задачи цифровой обработки изображений с использованием нейронных сетей [2, 3] стали относительно недавно. Это одна из причин того, что существует достаточно мало нейросетевых методов и алгоритмов для работы с фотографическими документами. Стоит обратить внимание на свёрточные и глубинные свёрточные сети. Они достаточно сильно отличаются от других существующих видов и обычно используются для обработки изображений. Одна из наиболее часто решаемых задач, которую они решают – классификация изображений. Однако, для восстановления фотографических документов стоит обратить внимание на следующий вид нейросетей. Это генеративно-состязательные сети (Generative Adversarial Network – GAN)

[2–5], которые используются для синтеза изображений без участия человека. Они состоят из двух нейронных сетей. Задача одной из них заключается в генерации данных, другой – отличать полученный результат от реального и оценить вероятность получения настоящих данных, а не сгенерированных. Первая сеть получила название генератор, вторая – дискриминатор.

Задача восстановления фотографических документов с использованием GAN заключается в обучении классифицировать, детектировать и генерировать фотографии, которое удовлетворяет требуемым результатам. Данную задачу лучше всего реализуют свёрточные генеративно-состязательные сети (CGAN) [2–5]. Их особенность, которая является огромным преимуществом при выборе данного вида нейросетей, заключается в том, что они обучаются генерировать искусственные изображения статистически неотличимые от реальных.

Процесс обучения состоит из следующих этапов. Из датасета выбирается пара фотографий – специально испорченное, которое поступит на вход генератору, и хорошее, с которым дискриминатор будет сравнивать результат работы первой сети. Сеть-генератор получает на вход точку (случайный вектор) из скрытого пространства, в дальнейшем переводит его в искусственное изображение. Сгенерированный объект получает сеть-дискриминатор и пытается определить его реалистичность. В процессе обучения происходит стабилизация точек скрытого пространства таким образом, что картинки, которые будут получены из точек одной окрестности, будут обладать рядом общих признаков. Это позволит поставить в соответствие каждому входному изображению некоторую точку, которая будет являться центром окрестности.

Один из ключевых моментов качественного процесса работы с нейросетью заключается в следующем. Нельзя допускать, чтобы в самом начале обучения дискриминатор побеждал слишком часто. В противном случае это может привести к остановке обучения. Данный момент решается добавлением внутренних циклов для сети-дискриминатора и сети-генератора, отслеживанием момента выхода, когда одна сеть почти догнала другую.

При решении задач восстановления цифровых изображений необходимо обратить внимание на различные методы и подходы, т.к. одного единого полноценного алгоритма не существует. К примеру, улучшение качества изображений реализуется генерацией сложного отображения между снимками низкого и высокого разрешений. Устранение шумов, в частности часто встречаемых импульсных и аддитивных Гауссовых, предполагает применение адаптивных алгоритмов или фильтров. Чаще всего используют медианные фильтры, получившие широкое применение в области цифровой обработки.

Следующий важный этап, который следует после выбора технологии для решения задачи, выбор языка программирования для её реализации.

3. Python и его программные библиотеки

При выборе языка программирования, который будет использован при разработке и реализации решения задачи восстановления фотографических документов, стоит обратить внимание на Python. Главными его преимуществами перед другими языками программирования является мощный механизм, который открывает доступ к быстрым вычислениям, лаконичность и выразительность, которые позволяют с минимальными затратами времени и сил работать с сложными алгоритмами.

Python [6] – один из высокоуровневых языков программирования, который ориентирован на повышение производительности разработчика и читаемости кода. Одним из его преимуществ является широкое применение и эффективность при решении задач машинного обучения, в частности с использованием нейронных сетей. В настоящее время создано огромное количество мощных библиотек, которые предназначены для работы с нейросетевыми проектами. Это и TensorFlow, которая помогает создавать сети со многими слоями, облегчает построение моделей глубокого обучения, позволяет получить доступ к хранилищу предва-

рительно обучаемых моделей. Для обучения нейронных сетей нельзя не упомянуть Keras, который упрощает работу с изображениями, позволяет вычислять функции потерь, определять процентную точность результата и создавать пользовательские функциональные слои.

4. Пример реализации

Одним из возможных примеров реализации данной можно назвать создание приложения, которое позволит пользователю загружать изображение для восстановления и в результате получить обработанный объект. На рис. 1 показана диаграмма вариантов использования.

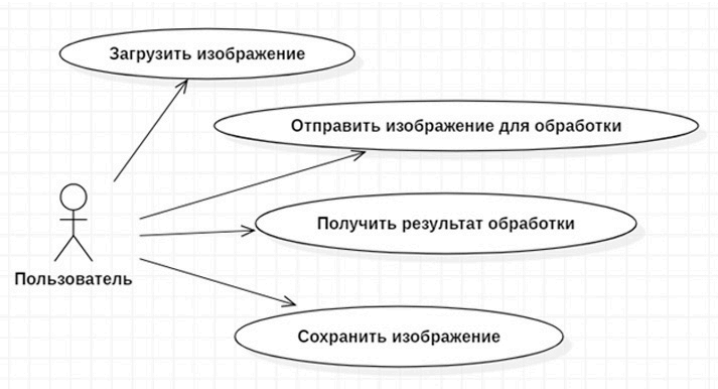


Рис. 1. Диаграмма вариантов использования приложения

Для создания приложения под платформу Android на языке программирования Python удобно использовать фреймворк Kivy [7]. Он поддерживает ускорение GPU своей графики, реализуемый интерфейс получается понятный. Главная идея данного фреймворка – лёгкость и быстрота приспособления пользователя к использованию программного обеспечения без чтения дополнительных инструкций. Пример того, как может выглядеть рабочая область приложения, которое было разработано на языке Python вместе с Kivy, продемонстрирован на рис. 2.

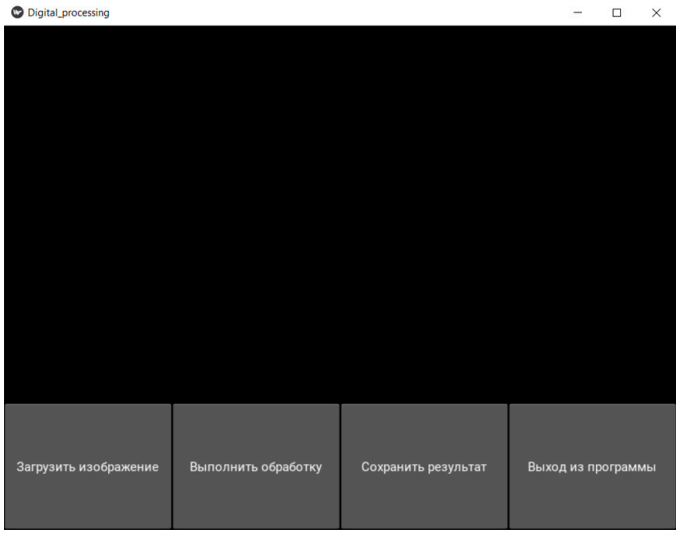


Рис. 2. Макет рабочей области

Все задачи восстановления повреждённых изображений разделены по сложности реализации. В зависимости от рассматриваемой модели шума, устранение шумов можно классифицировать как лёгкий процесс обработки, так и сложный. Ранее был упомянут медианный фильтр, который используется при решении данной цели. Его суть заключается в следующем. В функ-

цию, которая отвечает за обработку имеющегося шума на фотографии, поступает графический объект. Предполагается, что данное изображение $I1$ зашумлено и на выходе необходимо получить обработанное $I2$. Ключевыми параметрами данного алгоритма являются его параметр k (центр вариационного ряда, который делает данный алгоритм медианным) и некоторое небольшое нечётное число N, обязательно строго больше 1. Изображение рассматривается по точкам с координатами (i, j). Вокруг неё рассматривается окрестность $N \times N$, по значениям точек которой строится вариационный ряд p размера $N \times N$. Далее вычисляется значение точки нового изображения по правилу: если значение $| p(k) - I1(i, j) |$ строго меньше $| (N \times N - k + 1) - I1(i, j) |$, то точка с координатами (i, j) в изображении $I2$ примет значение $p(k)$, иначе – $p(N \times N - k + 1)$. Пример работы данного алгоритма изображён на рис. 3. Данный этап называется предобработкой изображения, которое на следующих этапах будет обработано с использованием нейронных сетей.

Рис. 3. Пример работы медианного фильтра

Улучшение качества изображений – это задача, которую можно отнести к менее сложным в реализации моментам. Методы, которые направлены на решение данной цели, называются super-resolution [2, 3]. Линейная фильтрация, бикубическая фильтрация, фильтрация Ланцоша. Данные алгоритмы продемонстрировали свою эффективность и быстроту, однако решения, полученные с их помощью, имеют чрезмерно гладкие текстуры. Итоговое изображение, максимально приближенное к реалистичному и правдоподобному, можно получить используя методы, которые полагаются на пару изображений с высоким и низким разрешением. Свёрточная генеративно-состязательная сеть получает на вход плохую картинку, которую изучает. Генератор пытается понять – какие изменения необходимо произвести, чтобы дискриминатор не смог определить – какое из двух изображений (сгенерированное и хорошее) подлинное. При создании сетей используется оптимизатор Adam, дискриминатор и генератор состоят из трёх скрытых слоёв с активационной функцией Leaky ReLU. Важный момент. Функция активации один из ключевых необходимых элементов, который оказывает влияние на обучаемость нейронной сети. Для надёжности сети-дискриминатора, в него добавляют dropout слои. Процесс генерации правдоподобной картинки происходит эпохами. Пример пары изображений высокого и низкого качества представлен на рис. 4, процесс создания новой картинки показан на рис. 5.

На первых этапах заметно, что фотографии не имеют реальной структуры и далеки от того, что должно получиться. Примерно на 40–50 эпохах появляются очертания и контуры картинки, постепенно изображение принимает нужную форму. Спустя 400 эпох мы можем наблюдать изображение, которое будет лучше, чем худшая картинка пары, и хуже, чем лучшая, которое показано на рис. 6.

Рис. 4. Пример пары LR-HR

Рис. 5. Процесс генерации изображения

Рис. 6. Результат работы спустя 400 эпох

Заключение

Задач и целей, озвученных в проблеме восстановления повреждённых фотографических документах, достаточно много. Как и методов, и подходов для их решения. Наибольшую эффективность по временным и трудовым затратам показывает использование нейронных сетей, которые реализуются с использованием высокоуровневого языка программирования Python. Изучение и исследование данной темы в настоящее время популярно и актуально. Данное исследование направлено на изучение основных моментов, нюансов и тонкостей при использовании упомянутых технологий. Улучшение алгоритмов и процессов обучения нейросетей в самом начале развития.

Литература

1. Обработка изображений. – Режим доступа: https://ru.wikipedia.org/wiki/Обработка_изображений (дата обращения: 6.04.2021).

2. *Будума, Н.* Основы глубокого обучения / Н. Будума. – МИФ Бизнес, 2017. – 304 с.

3. Шолле, Ф. Глубокое обучение на Python / Ф. Шолле. – Санкт-Петербург : Питер, 2018 – 400 с.

4. Генеративно-состязательная нейросеть (GAN). Руководство для новичков. – Режим доступа: https://neurohive.io/ru/osnovy-data-science/gan-rukovodstvo-dlja-novichkov (дата обращения: 12.04.2021).

5. Generative Adversarial Nets (GAN). – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=Generative_Adversarial_Nets_(GAN) (дата обращения: 15.04.2021).

6. Официальный сайт Python. – Режим доступа: https://www.python.org/ (дата обращения: 29.03.2021).

7. Официальный сайт Kivy. – Режим доступа: https://kivy.org/#home (дата обращения: 30.03.2021).

Воронова Ольга Сергеевна – студентка 4-го курса кафедры МО ЭВМ Воронежского государственного университета. E-mail: deer-laziness@mail.ru

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры МО ЭВМ Воронежского государственного университета.
E-mail: bolotova.svetlana@gmail.com

УДК 004.925

ПОСТРОЕНИЕ 3D МОДЕЛИ ХРЯЩА КОЛЕННОГО СУСТАВА ПО СНИМКАМ МРТ, МЕТОДОМ СПЛАЙНОВОЙ ИНТЕРПОЛЯЦИИ

А. Ю. Горяинова, Е. В. Трофименко

Воронежский государственный университет

Введение

В настоящее время одним из наиболее тяжёлых и распространенных заболеваний опорно-двигательного аппарата является остеоартроз (ОА). Это дегенеративно-дистрофическое заболевание суставов, причиной которого является поражение хрящевой ткани суставных поверхностей[1]. Смещение между двумя костями вызывает у пациента боль, дискомфорт в области трения, ограниченное или затрудненное движение. Эта ситуация касается не только спортсменов или активных людей, обычно это может случиться с кем угодно, независимо от пола, возраста или происхождения. Именно поэтому так важно диагностировать проявление болезни на ранних стадиях развития. Поскольку магнитно-резонансная томография (МРТ) может напрямую визуализировать суставной хрящ, она используется в медицинской диагностике для измерения линейных размеров и количественной оценке тканей. При помощи сегментации можно получить подробную информацию о состоянии пораженной области и подобрать соответствующее лечение. Однако сегментация изображений коленного хряща, является очень сложной задачей из-за тонкой структуры, а также синовиальной жидкости или других окружающих тканей. Поэтому данная задача является крайне актуальной в настоящее время.

1. Постановка задачи

Необходимо разработать модуль, реализующий алгоритм построения интерполяционной поверхности по замкнутым контурам, которые будут строиться на основе DICOM снимков, полученных с аппарата магнитно-резонансной томографии (МРТ). Проанализировать полученные данные и оценить использование данного метода сегментации при построении 3D модели хряща.

2. Обзор существующих подходов сегментации

Необходимость сегментации участков на изображении и 3D моделях в медицине появилась довольно давно и существует немало подходов, успешно решающих данную задачу. Однако каждый метод имеет свои преимущества и недостатки. Опишем некоторые из них.

2.1. Алгоритм обработки сагиттальных изображений коленного сустава

Предварительная обработка для уменьшения белого шума включала применение фильтра Винера с размером маски 3×3, не снижающей пространственное разрешение. Сегментация хряща осуществлялась методом бинаризации с двойным ограничением и последующим применением морфологических операций (наращивание, эрозия, устранение разрывов и др.) с числом итераций не более 5, что связано с малой толщиной хряща. Распознавание объектов и устранение ошибок сегментации проводилось на основе метода поиска объектов по площади.

Выделение костных структур выполнялось на T1 FSE изображениях с использованием медианного фильтра с маской 5×5, преобразованием гистограммы и двухпороговой сегментацией, объединенной с методом выделения границ Канни [3].

2.2. Полуавтоматическая сегментация с использованием метода на основе обнаружения краев и сплайнов Безье

Хрящ был полуавтоматически сегментирован на сагиттальных SPGR изображениях на основе обнаружения краев и сплайнов Безье. Итеративный процесс минимизации был использован для расчета общего объема хряща и средней толщины для каждого региона. После сегментации медиальная линия была сформирована в каждой области хряща. Толщина хряща определялась путем расчета минимального расстояния от каждой точки на средней линии до границы хряща. Средняя толщина рассчитывалась для каждого среза, а затем усреднялась для всех срезов. Объем хряща определялся путем умножения общего количества вокселей, охватывающих хрящ, на объем каждого вокселя. Наконец, чтобы минимизировать объемные изменения из-за размера коленного сустава, объем хряща нормализовывался по эпикондилярному расстоянию, определенному по осевым изображениям SPGR [4].

2.3. Watersheds transform

В данном подходе с помощью анизотропной фильтрации сглаживается изображение с последующей ручной аннотацией трех типов маркеров: кости, хрящи и другие ткани (связки, мышцы и т. д.). Используя данные маркеры в качестве локальных минимумов, алгоритм пытается найти пересечения так называемых водосборных бассейнов, областей, которые находятся выше местного минимума. Проблема избыточной сегментации, которая характерна для метода водораздела, решается путем применения геодезической реконструкции и оптимизируется с использованием упорядоченных очередей. Вместо использования градиентного изображения в качестве входных данных для преобразования водораздела вводится функция, которая вычисляет вероятность нахождения грани между помеченными и немечеными вокселями. Интенсивности между различными тканями имеют нормальное распределение и по своей природе демонстрируют пространственную однородность, что фактически характерно для изображений, используемых при валидации.

Алгоритм был проверен на множестве ручных сегментаций, определенных двумя экспертами независимо друг от друга. Точность полуавтоматической сегментации оказалась такой же хорошей, как и ручная сегментация. На этапе инициализации нужно было определить 50 маркеров для каждого класса тканей, что занимало в среднем 5–10 минут на набор данных. Хотя это значительно ниже, чем сегментирование изображений вручную (2 часа на набор данных). Однако следует иметь в виду, что геодезическая реконструкция сгладит все водосборные бассейны, которые не были отмечены, поэтому важно учитывать местоположение при размещении маркеров [5].

Так же существуют не только полуавтоматические, но и полностью автоматические методы сегментации. К одному из таких относится следующий алгоритм.

2.4. Supervised learning

Данный метод основан на kNN подходе к сегментации хряща. Алгоритм состоит из двух разных бинарных классификаторов, отдельно для большеберцового и бедренного хрящей. Классификаторы были настроены так, чтобы включать особенности, специфичные для типа хряща, который сегментируется.

Для большеберцового хряща были включены следующие особенности: положение на изображении, интенсивность после сглаживания с помощью гауссовского фильтра (три различных размера ядра), производные первого порядка, собственные значения гессианы и собственные значения тензора структуры, генерируемые из производных первого порядка, свернутых с гауссовским [6].

3. Алгоритм

Для сегментации использовалась DICOM серия с T2-картированием. Это проверенная и надежная техника, характеризующая структуру и пространственную ориентацию коллагена, и количество окружающей воды. Здесь хорошо заметен хрящ, что упрощает работу по выделению границ.

Алгоритм работы программы для сегментации хряща состоит в следующем:

1. Выбрать DICOM серию с коленным суставом.
2. Расставить точки на границе хряща.
3. Повторить п.2 для каждого снимка.
4. После получения модели (если необходимо) отредактировать неточности, путем сдвига опорных точек.

Для данного алгоритма использовались опорные точки, которые пользователь расставлял самостоятельно. Затем по ним строилась интерполяция B-сплайном, в результате чего получалась замкнутая линия[2]. Когда это проделано для всех снимков, строится поверхность по полученным контурам. Это реализовано посредством библиотеки Open Cascade. Программа была реализована на языке C++ с использованием фреймворка Qt. Так же в качестве ядра программы использовался продукт Open Cascade Technology. Он сочетает в себе набор библиотек и средств разработки программного обеспечения, ориентированного на 3D-моделирование, в особенности систем автоматизированного проектирования.

4. Результаты

В данной работе была построена 3D модель сегментированного хряща коленного сустава путем построения интерполяционной поверхности по замкнутым контурам, опираясь на снимки МРТ.

На рис. 1 расставлены опорные точки, изображенные в желтом цвете, которые расположены на границах хряща. По ним построен сплайн красного цвета. Такая процедура проводилась для всех снимков из серии. Путем интерполяции по контурам получается замкнутая фигура. Результатом работы получается 3D модель сегментированного хряща, представленная на рис. 2.

Данный метод представляет собой полуавтоматическую сегментацию и является довольно затратным как по времени, так и по силам, которые необходимо приложить для точного позиционирования точек. Если целостность хряща нарушена, то есть отсутствуют ткани на некоторых участках, то данный алгоритм непригоден. Так же, если набор снимков слишком большой, а сегментируемая поверхность слишком неоднородная и нуждается в большом количестве точек, то работа может занять довольно много времени. При необходимости изменить форму полученной трехмерной модели требуется время для пересчета. Вследствие этого можно сделать вывод, что алгоритм нуждается в существенной доработке и дополнительной оптимизации.

Заключение

В настоящей работе был предложен метод сегментации хряща коленного сустава путем построения сплайновой поверхности по замкнутым контурам. Так как результаты его рабо-

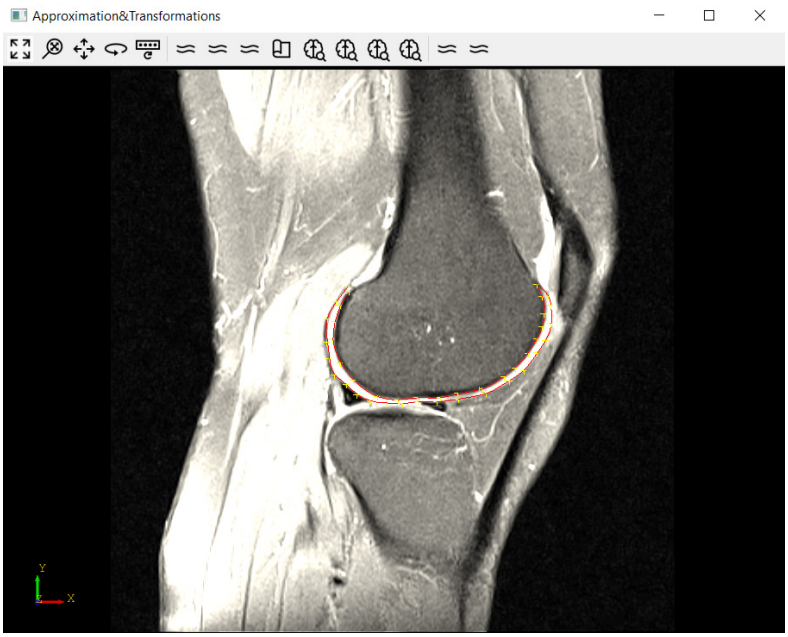


Рис. 1. Установка опорных точек на границах хряща и построение сплайна

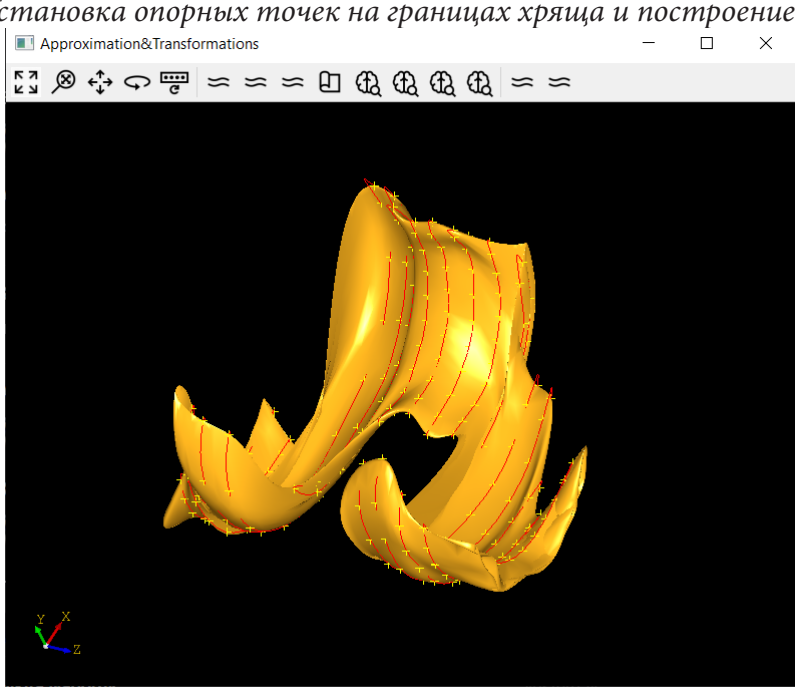


Рис. 2. Сегментированная 3D модель хряща коленного сустава

ты имеют довольно большую погрешность, а работа для установки опорных точек занимает довольно много времени, то можно сделать вывод, что он не пригоден для данной задачи. Для столь небольших объектов, как хрящ нужны более точные методы, позволяющие учесть величину и неоднородность области.

Литература

1. *Покровский В. И.* Малая медицинская энциклопедия / Советская энциклопедия, 1996. – Т. 4. – 577 с. – ISBN 5-225-02819-5.

2. *Роджерс Д.* Математические основы машинной графики / Роджерс Д., Адамс Дж. – Пер. с англ. – М. : Мир, 2001. – 604 с.

3. *Анотонова А. С., Казначеева А. О.* Морфологический анализ в задачах автоматизации обработки изображений коленного сустава // Изв. вузов. Приборостроение. – 2018. – Т. 61, № 2. – С. 186–191.

4. *Li X., Benjamin Ma C., Link T. M., Castillo D.-D., Blumenkrantz G., Lozano J., Carballido-Gamio J., Ries M., Majumdar S.* In vivo T1p and T2 mapping of articular cartilage in osteoarthritis of the knee using 3T MRI // International Cartilage Repair Society. OsteoArthritis and Cartilage. – 2007. – No 15. – P. 789–797.

5. *Grau Vicente, Mewes AUJ, Alcaniz M, Kikinis Ron, Warfield Simon K.* Improved watershed transform for medical image segmentation using prior information // Medical Imaging, IEEE Transactions on. – 2004. – 23(4). – P. 447–458.

6. *Folkesson Jenny, Dam Erik B, Olsen Ole Fogh, Pettersen Paola C, Christiansen Claus.* Segmenting articular cartilage automatically using a voxel classification approach // Medical Imaging, IEEE Transactions on. – 2007. – 26(1). – P. 106–115.

Горяинова Алина Юрьевна – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: goryainova.9745@gmail.com

Трофименко Елена Владимировна (научный руководитель) – канд. физ.-мат. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: evtrof@gmail.com

УДК 658.7

ПРОБЛЕМЫ МАРШРУТИЗАЦИИ СБОРЩИКА ПРИ КОМПЛЕКТАЦИИ ЗАКАЗОВ В УСЛОВИЯХ РАСПРЕДЕЛИТЕЛЬНОГО ЦЕНТРА: АНАЛИТИЧЕСКИЙ ОБЗОР ЗАДАЧ И МЕТОДОВ ИХ РЕШЕНИЯ

О. Г. Духанина

Воронежский государственный университет

Каждый год число крупных предприятий растет, так же растет и их масштаб: предприятия открывают новые заводы, цеха, склады. Преимущественно, предприятия большого масштаба являются производственными – основная сфера деятельности это создание и выпуск продукции. Для осуществления основных сфер деятельности необходимо поддерживать такие процессы, как закупка, планирование, сбыт, финансово-учетный процесс, управление персоналом и складской деятельностью.

В рамках логистической цепочки за функцию складирования, распределения и отгрузки продукции отвечает склад. За последние годы функция склада изменилась: если раньше склад предназначался только для хранения, то сейчас функции склада расширены до контроля качества, сборки, группировки, реверсивной логистики, систематизации, комплектовании товаров.

Задача маршрутизации сборщика при комплектации заказов на складе возникает на крупном предприятии с большим складом или складском комплексе. На складах такого масштаба используется особая система хранения, где в процессах склада участвует не только персонал, но и мобильные комплектовщики заказов.

Из функций склада особо следует выделить процедуру комплектования. Комплектование или сборка заказов представляет собой сбор товара из определенных мест хранения, ячеек, и отправки товара на отгрузку. Процесс комплектования (сборки) заказов является одним из приоритетных направлений для усовершенствования процедуры складирования, так как данный процесс занимает больше времени относительно других складских задач [3].

Для процесса комплектации заказов характерны такие аспекты как:

– поставка в пункт сборки требуемого объема продукции;

– накопление определенной продукции в соответствии с персональными требованиями заказчика;

– сортировка продукции в соответствии с персональными требованиями заказчика.

Одним из ключевых процессов на складском предприятии является передача информации. В этот процесс входит передача информации об инвентаризации, координатах ячеек хранения, объемах партий, информация о количестве товара, информация и описание.

Кластеризация и планирование, выборка товаров из ячеек хранения, отправка собранных товаров на отгрузку – все эти и многие другие процессы входят в задачу комплектования заказов.

На сегодняшний день имеется большое количество различных систем комплектования заказов, которые могут функционировать на складе одновременно. На рис. 1, который представлен ниже, представлены виды систем комплектования заказов.

Существуют системы механические, автоматические, а также системы с ручной сборкой [2].

Из перечисленных выше систем стоит выделить автоматические системы комплектации заказов.

Автоматические системы осуществляют быстрый и эффективный подбор заказов, состоящих из нескольких позиций продуктов. Концепция этих систем подбора заказов также предус-

Рис. 1. Виды систем для комплектации заказов

матривает постоянное отслеживание товаров. Автоматические системы отличаются высокой надежностью, скоростью и возможностью комбинировать автоматические системы с другими системами хранения и складирования.

Цели внедрения автоматической системы:

– активное управление складом;

– получение точной информации о месте нахождения товара на складе;

– эффективное управление товаром, имеющим ограниченные сроки годности;

– получение инструмента для повышения эффективности и развития процессов по обработке товара на складе;

– оптимизация использования складских площадей.

Для решения задач маршрутизации используются эвристические методы маршрутизации. К данным методам относятся:

– *S*-образный метод;

– метод с возвратами;

– серединный метод;

– последовательный метод;

– составной метод.

S-образный метод основан на том, что любой проход, содержащий хотя бы одну сборочную ячейку должен быть пересечен полностью от ближнего (дальнего) поперечного прохода до дальнего (ближнего) поперечного прохода. Пример применения *S*-образной эвристики на типовом заказе представлен на рис. 2. Данной эвристике посвящены работы [6].

При использовании метода с возвратами сборщик заказов заходит и выходит из сборочных проходов через один поперечный проход, пересекая сборочный проход целиком только в случае перехода из одного блока в другой. Такое пересечение возможно сделать только либо через крайний левый под-проход с не посещенными сборочными ячейками, либо через крайний правый под-проход с не посещенными сборочными ячейками. Данный эвристический метод маршрутизации рассматривается в работах [5]. Схема построения маршрута по методу с возвратами указана на рис. 3.

Серединный метод маршрутизации фактически разбивает каждый блок склада на две секции. Сборочные ячейки, относящиеся к ближней (к базе) секции блока, посещаются через ближний поперечный проход блока; сборочные ячейки дальней секции – через дальний поперечный проход. Сборщик заказов пересекает проход целиком только для перехода из одного блока в другой, при этом это возможно сделать только через крайний левый под-проход с не посещенными сборочными ячейками, либо через крайний правый под-проход с не посещенными сборочными ячейками. Данный эвристический метод маршрутизации представлен в работах [4].Схема построения маршрута при использовании серединного метода указана на рис. 4.

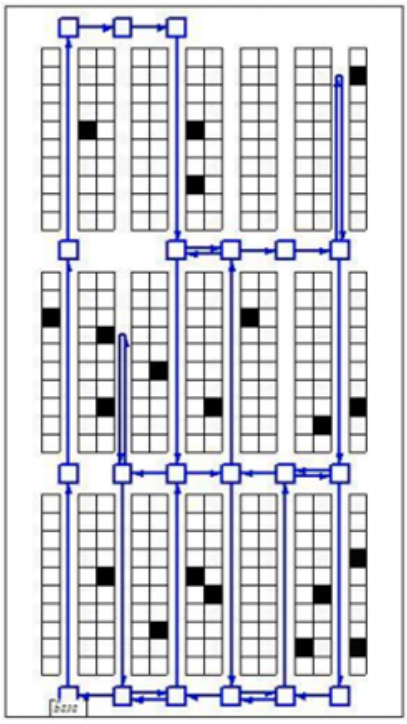


Рис. 2. S-образный метод

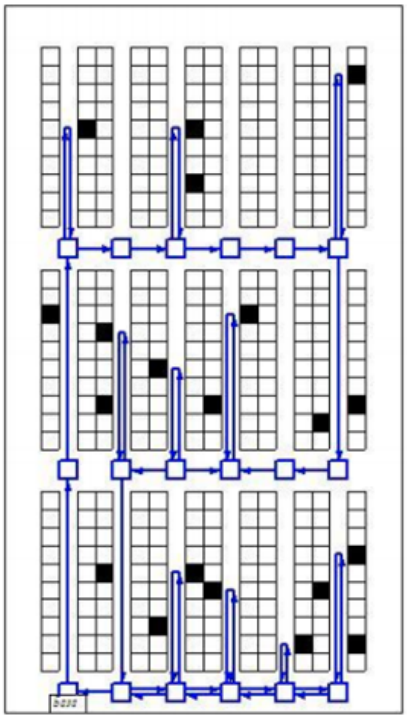


Рис. 3. Метод с возвратами

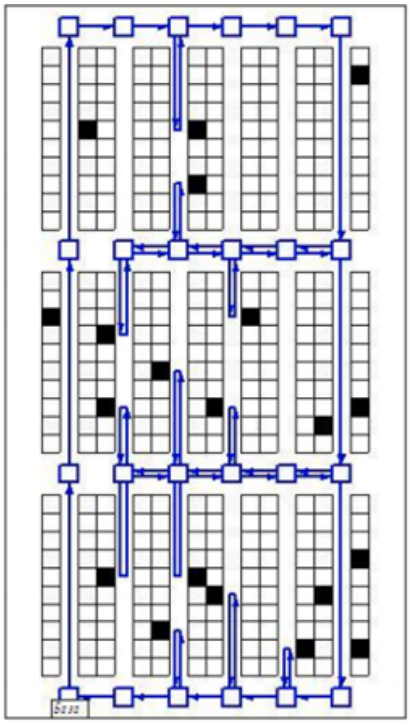


Рис. 4. Серединный метод

Последовательный эвристический метод маршрутизации в многоблочных складах представлен в работе [7]. Сборочный маршрут, получающийся в результате использования данной эвристики, посещает каждый сборочный проход только один раз. Таким образом, вначале собираются все необходимые артикулы из первого прохода, затем – из второго и т. д. Для определения поперечного прохода для перехода между сборочными проходами используется динамическое программирование.

При использовании последовательного метода находится значение $C_m(i, j)$. Значение $C_m(i, j)$ это расстояние сборочного прохода m, которое требуется пройти, стартовав из i-го поперечного прохода (на первой итерации, из ближайшего к базе ближнего поперечного прохода), собрав все необходимые артикулы в первом проходе и перейдя на следующий сборочный проход через j-й поперечный проход, следующим образом:

$$C_m(i, j) = B_{1m}(i, j) + |i - j|(L + A) + B_{2m}(i, j),$$

где $B_{1m}(i, j)$ – длина плеча для обхода всех сборочных ячеек прохода m, находящихся дальше поперечного прохода $\min(i, j)$;

$B_{2m}(i, j)$ – длина плеча для обхода всех сборочных ячеек прохода m, находящихся ближе поперечного прохода $\max(i, j)$;

L – длина блока склада (без учета поперечных проходов);

A – ширина поперечных проходов.

Величина $B_{1m}(i, j)$ определяется по формуле:

$$B_{1m}(i, j) = \begin{cases} 0, \text{ если } K_m = 0 \text{ или } X_m^- \geq \min(iL, jL); \\ 2\left[\min(iL, jL) - X_m^- + A\left(0.5 + \left\lfloor \frac{\min(iL, jL) - X_m^-}{L} \right\rfloor\right)\right] \text{ в противном случае, где} \end{cases}$$

K_m – число сборочных ячеек прохода m;

$X_m^- = \min_t \{X_m(t)\}$ – продольная координата наиболее удаленной (от базы) сборочной ячейки прохода m;

iL, jL – длина блока склада с учетом проходов;

$X_m(t)$ – продольная координата сборочной ячейки t прохода $m\left(0 \le X_m(T) < T\right)$, $m = 1, 2, ..., M$, $t = 1, 2, ..., K_m$.

Величина $B_{2m}(i, j)$ определяется по формуле

$$B_{2m}(i,j) = \begin{cases} 0, \text{ если } K_m = 0 \text{ или } X_m^+ < \max(iL, jL) \\ 2\left[\max(iL, jL) - X_m^+ + A\left(0.5 + \left\lfloor \dfrac{X_m^+ + \max(iL, jL)}{L} \right\rfloor\right)\right] \text{ в противном случае.} \end{cases}$$

Здесь $X_m^+ = \max_t \{X_m(t)\}$ – продольная координата ближайшей (к базе) сборочной ячейки прохода m.

Пример применения последовательной эвристики на типовом заказе представлен на рис. 5.

Составной метод предложен в работе [5]. Он сочетает в себе преимущества S-образной эвристики и эвристики с возвратами. Метод минимизирует преодолеваемое расстояние между наиболее удаленными сборочными ячейками двух смежных под-проходов с не посещенными сборочными ячейками и определяет наилучший способ преодоления прохода – пересечение или заход с возвратом [1]. На рис. 6 представлен пример применения составного метода.

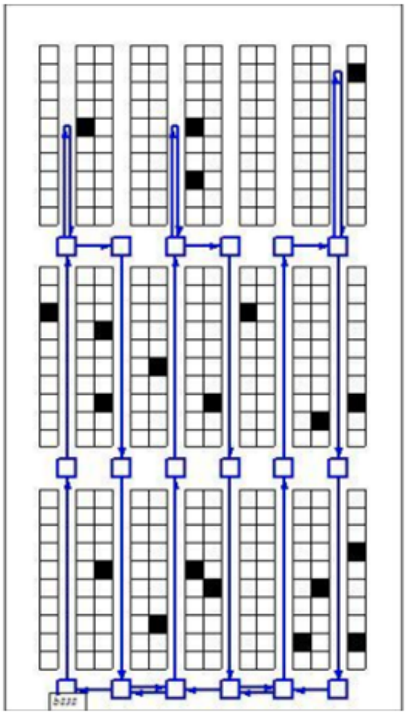


Рис. 5. Последовательный метод

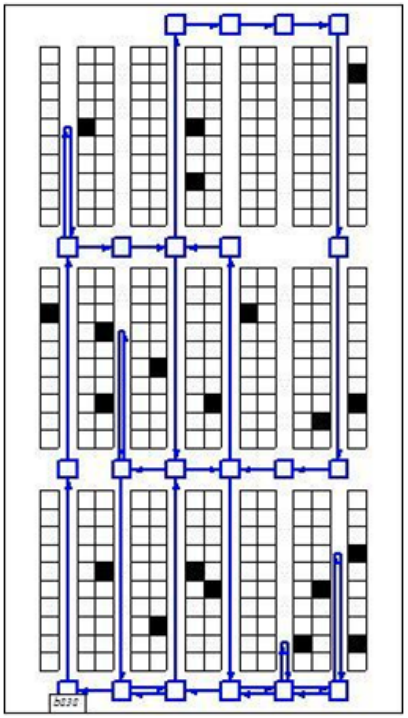


Рис. 6. Составной метод

На основании анализа приведенных методов можно сделать вывод, что каждый из них направлен на оптимизацию процесса сборки заказов в зависимости от условий прохождения маршрута. В рамках большого предприятия, где складские процессы являются приоритетными, организация процесса комплектации тесно связана с процессами производства готовой продукции и планированием. В таких случаях многие производства внедряют систему MES. Данная система предназначена для решения задач синхронизации, координации, анализа и оптимизации выпуска продукции в рамках какого-либо производства. MES-системы относят-

ся к классу систем управления уровня цеха, но могут использоваться и для интегрированного управления производством на предприятии в целом. Внедрение системы MES в производство приведет к оптимизации процесса комплектации заказов, увеличит скорость складских процессов, а также обеспечит контроль несоответствий данных системы и физических данных.

Литература

1. *Коробков Е. В.* Процесс комплектования заказов на складе. Обзор / Е. В. Коробков // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. – 2015. – № 3. – С. 153–183.

2. *Гаджинский А. М.* Логистика: Учебник для высших и средних специальных учебных заведений. – Москва: Маркетинг, 1999. – 27 с.

3. *Bartholdi III J., Hackman S. T.* Warehouse and Distribution Science Release 0.94. Atlanta, GA, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering, Georgia Institute of Technology. – 2011. – 300 p.

4. *De Koster R., Le-Duc T., Roodbergen K. J.* Design and Control of Warehouse Order Picking: A Literature Review // European Journal of Operational Research. – 2007. – V. 182, No. 2. – P. 481–501.

5. *Petersen II C. G.* An Evaluation of Order Picking Routeing Policies // International Journal of Operations and Production Management. – 1997. – V. 17, No. 11. – P. 1098–1111.

6. *Ratliff H. D., Rosenthal A. S.* Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem // Operations Research. – 1983. – V. 31, No. 3. – P. 507–521.

7. *Vaughan T. S.* The Effect of Warehouse Cross Aisles on Order Picking Efficiency // International Journal of Production Research. – 1999. – V. 37, No. 4. – P. 881–897.

Духанина Ольга Геннадьевна – магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
Email: olg.duk@yandex.ru

Булгакова Ирина Николаевна (научный руководитель) – д-р экон. наук, доц., доцент кафедры системного анализа и управления Воронежского государственного университета. Email: bulgakova-i-n@yandex.ru

УДК 004.93’11

АЛГОРИТМЫ РАСПОЗНАВАНИЯ ЛИЦ

В. А. Енокян

Воронежский государственный университет

Введение

Биометрия относится к автоматической идентификации человека на основе его или ее физиологических, или поведенческих характеристик. Биометрическая система – это система распознавания образов, которая работает, собирая данные от человека и сравнивая их с набором шаблонов, хранящимся в базе данных. Под «биометрикой» подразумеваются физические характеристики человека, такие как лицо, геометрия руки, отпечатки пальцев, голос, ладонь, подпись и радужная оболочка. Как метод идентификации, распознавание лиц предпочтительнее традиционных методов, включающих пароли и ПИН-коды по нескольким причинам, среди которых – необходимость физического присутствия лица, подлежащего идентификации, и/или идентификации, основанной на биометрических методах, что устраняет необходимость запоминать пароль.

1. Распознавание лиц

Распознавание лиц относится к наиболее актуальным приложениям анализа изображений. Данная задача включает в себя распознавание образов и обработку изображений. Можно выделить два типа процедур, которые лежат в основе распознавания:

– проверка, когда система сравнивает данного человека с человеком, которым он себя называет, и принимает решение «да» или «нет»;

– идентификация, когда система сравнивает данного человека со всеми другими людьми, хранящимися в базе данных, и дает ранжированный (упорядоченный) список совпадений.

Методы распознавания лиц включают в себя совокупность шагов, которые заключаются в захвате, анализе и сравнении заданного лица с базой данных сохраненных изображений [7]. Ниже представлен базовый процесс, который используется системой распознавания:

Шаг 1. *Обнаружение лица*.

Программа распознавания ищет лицо с помощью видеокамеры, когда система подключена к системе видеонаблюдения;

Шаг 2. *Выравнивание*.

Как только система обнаруживает лицо, она определяет положение, размер и позу головы. Чтобы система зафиксировала лицо, его необходимо повернуть не менее чем на 35 градусов в сторону камеры.

Шаг 3. *Нормализация*.

Чтобы изображение головы было зарегистрировано и сопоставлено с соответствующими размерами и позой, оно масштабируется и поворачивается. Нормализация выполняется независимо от расположения головы и расстояния от камеры;

Шаг 4. *Представление*.

После выполнения нормализации система преобразует данные лица в уникальный код. Этот процесс кодирования позволяет упростить представление и сравнение вновь полученных данных лица с данными лица, которые уже сохранены.

Шаг 5. *Сравнение*.

Полученные данные о лице сравниваются с сохраненными данными и связываются по меньшей мере с одним сохраненным изображением лица. Система решает, совпадают ли черты лица, извлеченные из вновь полученных данных. Если результат превышает заранее установленный порог, объявляется совпадение.

Описанный процесс представлен на рис. 1.

Рис. 1. Базовый процесс системы распознавания лиц

2. Алгоритмы распознавания лиц

2.1. Метод главных компонент

Одним из наиболее известных и проработанных является метод главных компонент (principal component analysis, PCA), основанный на преобразовании Карунена – Лоева [1, 2, 4]. Первоначально метод главных компонент начал применяться в статистике для снижения пространства признаков без существенной потери информации. В задаче распознавания лиц его применяют главным образом для представления изображения лица вектором малой размерности (главных компонент), который сравнивается затем с эталонными векторами, заложенными в базу данных.

Описание метода:

1. Пусть имеется набор лиц для обучения $\{X_1, X_2, ..., X_M\}$, т. е. фотографий одного лица, представленного в разных ракурсах, вариантах прически, выражения лица и т. п. На основе заданного набора изображений определяется «среднее лицо»

$$X = \frac{1}{M}\sum_{i=1}^{M} X_i.$$

2. Для каждого изображения определить отклонение от среднего вектора признаков

$$\phi_i = \overline{X_i} - X = \overline{X_i} - \frac{1}{M}\sum_{i=1}^{M} X_i.$$

3. Вычислить матрицу ковариации

$$C = \frac{1}{M}\sum_{i=1}^{M} \phi_i \phi_i = A^T A,$$

где $A = [\phi_1, \phi_2, ..., \phi_M]$ – матрица размерности $N \times M$.

4. Вычислить собственные векторы и собственные значения ковариационной матрицы.

5. Выбор компонентов и формирование вектора признаков: упорядочить собственные векторы по убыванию собственных значений, тогда собственный вектор с максимальным собственным значением является основным компонентом набора данных.

6. Получение новых наборов данных: после выбора компонентов (собственных векторов), которые хотим сохранить в данные, сформировать вектор признаков.

К основным преимуществам применения анализа главных компонент относятся:

• хранение и поиск изображение в больших базах данных;

• возможность реконструкции изображений.

Основной недостаток – высокие требования к условиям съемки изображений. Изображения должны быть получены в близких условиях освещенности, одинаковом ракурсе и должна быть проведена качественная предварительная обработка, приводящая изображения к стандартным условиям (масштаб, поворот, центрирование, выравнивание яркости отсечение фона). Нежелательно наличие таких факторов, как очки, изменения в прическе, выражении лица и прочих внутриклассовых вариаций.

2.2. Метод геометрических характеристик

Метод геометрических характеристик лица является одним из самых первых и самых простых методов распознавания лиц [3, 4]. Данный метод основывается на определении ключевых точек лица с последующим выделением набора признаков, каждый из которых является либо расстоянием между ключевыми точками, либо отношением таких расстояний. В качестве ключевых точек можно взять уголки глаз, губ, кончик носа, центр глаза и т. д. (рис. 2). Ключевыми областями могут быть прямоугольные области, которые включают в себя: глаза, нос, рот. Данный метод предполагает строгие требования к условиям съемки, нуждается в надежном механизме поиска ключевых точек.

Описание метода:

1. Нормализация изображения.

Так как в данном методе используются фронтальные изображения лица, процесс нормализации сводится к задаче нахождение коэффициента масштабирования k.

В качестве способа нахождения коэффициента k рассмотрим следующий способ.

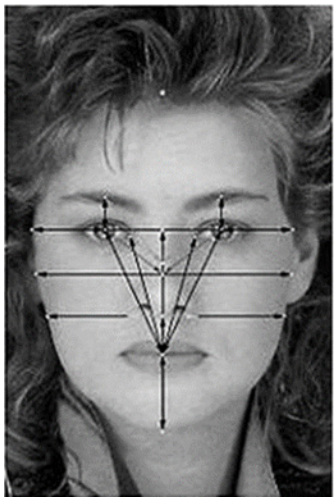


Рис. 2. Идентификационные точки и расстояния

Положим на этапе обучения определено стандартное расстояние L_0 от объекта до системы наблюдения. Пусть также на этапе идентификации мы можем знать фактическое расстояние L

от объекта до системы. Тогда неизвестный параметр нормализации определяется следующим образом:

$$k = L \big/ L_0 .$$

2. Нахождение векторов-признаков.

Определим следующие положения ключевых точек (рис. 3): 1-я точка – центр тяжести лица; 2, 3-я – центры тяжести правого и левого глаза соответственно; 4, 5, 6, 7-я – уголки глаз; 8-я – центр тяжести губ; 9, 10-я – уголки губ

Рис. 3. Положения ключевых точек

Центр тяжести односвязного контура определяется по формуле

$$x_c = \frac{1}{M_c}\sum_{i=1}^{M_c} x_i; \quad y_c = \frac{1}{M_c}\sum_{i=1}^{M_c} y_i,$$

где M_C – количество единичных точек контура.

Векторы-признаки (расстояние между ключевыми точками) вычисляются по формуле

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

3. Идентификация

Для поиска распознанного лица выполняется поиск по всей базе данных, используя одну фотографию или, для сужения диапазона поиска, используют дополнительную информацию о человеке (пол, возраст и т. д.), при этом, могут применяться следующие основные поля БД: ID – индекс фотографии; Name – имя человека, которому принадлежит фотография; Path – путь к каталогу, содержащему фотографии; FileName – имя файла, хранящего массивы признаков (рис. 4).

Рис. 4. Структура файла

Заголовок содержит информацию о размере файла, смещение признаков относительно начала основной части. В биометрических данных находятся массивы векторов-признаков для различных контуров лица.

Для того, чтобы соотнести распознанное изображение и изображение, находящееся в базе данных, вычисляют коэффициент корреляции для всех записей базы данных.

Процедура обработки информации включает следующие шаги:

1. Определяется отношение расстояний распознаваемого изображения и хранящегося в базе данных (либо же отношение площадей).

2. Если максимальные значения отношений не меньше заданного порога, то вычисляется коэффициент корреляции и запоминается ID фотографии в базе данных.

3. Находится коэффициент корреляции, наиболее близкий к 1, и выводится фотография с данным ID.

К преимуществам данного метода можно отнести простоту реализации, менее трудоёмкий математический аппарат. К недостаткам относятся строгие требования к условиям съёмки: хорошо освещенное нейтральное изображение лица без помех, сфотографированное фронтально.

2.3. Нейронные сети

Для достижения высокой точности распознавания нейронная сеть предобучается на большом массиве изображений, например, таком, как в базе данных MegaFace. Это основной метод обучения для распознавания лиц [5, 6].

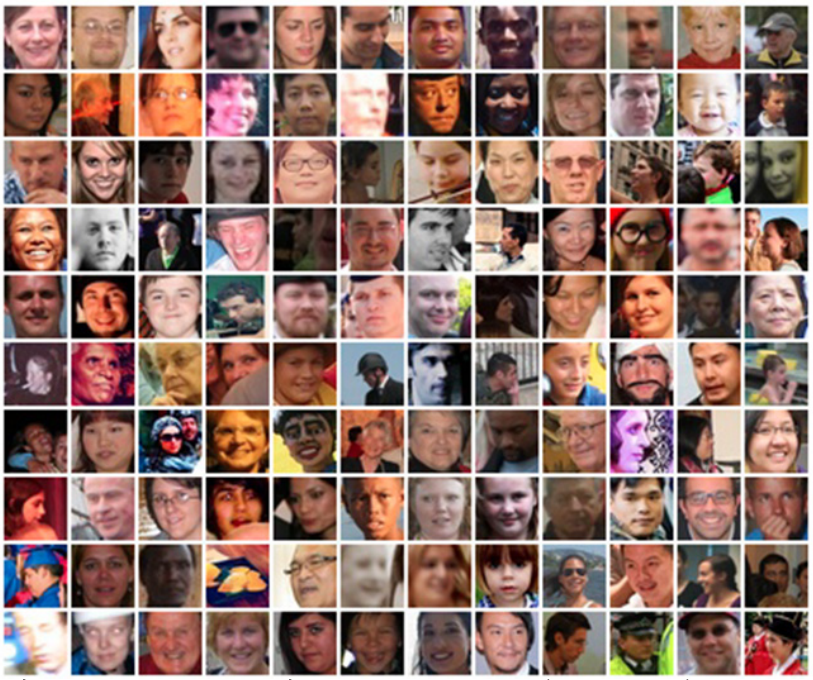


Рис. 5. База данных MegaFace содержит 1 млн. изображений более 690 тыс. людей

После того, как сеть обучена распознавать лица, процесс распознавания лица может быть описан следующим образом (рис. 6). Сначала изображение обрабатывается с помощью детектора лица: алгоритма, который определяет прямоугольный фрагмент изображения с лицом. Этот фрагмент нормализуется для того, чтобы легче обрабатываться нейронной сетью: наилучший результат будет достигнут, если все входные изображения будут одинакового размера, цветности и т. д. Нормализованное изображение подаётся на вход нейронной сети для обработки алгоритмом. Данный алгоритм обычно является уникальной разработкой компании

для повышения качества распознавания, однако существуют и «стандартные» решения для данной задачи. Нейронная сеть строит уникальный вектор признаков, который затем переносится в базу данных. Поисковая система сравнивает его со всеми векторами признаков, хранящихся в базе данных, и даёт результат поиска в виде определённого числа имён или профилей пользователей со схожими лицевыми признаками, каждому из которых присваивается определённое число. Это число представляет собой степень схожести нашего вектора признаков с найденным в базе данных.

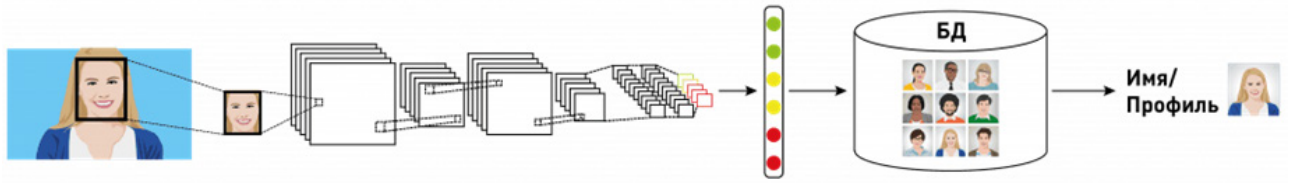


Рис. 6. Процесс распознавания лица

К недостаткам нейронных сетей относится то, что добавление нового эталонного лица в базу данных требует полного переобучения сети на всем имеющемся наборе (достаточно длительная процедура, в зависимости от размера выборки от 1 часа до нескольких дней).

3. Сравнительный анализ

Был проведен теоретический анализ рассмотренных алгоритмов. Результаты анализа приведен в табл. 1.

Таблица 1

Метод распознавания лиц	Требования к условиям съемки	Предварительная обработка изображений	Реализация	Точность распознавания	Влияние мимики на точность распознавания
Метод главных компонент	высокие	да	средняя	~ 90%	высокое
Метод геометрических характеристик	высокие	да	простая	< 90%	высокое
Нейронные сети	низкие	да	сложная	> 90%	низкое

Заключение

В данной статье были описаны некоторые из алгоритмов распознавания лиц. Были проанализированы основные проблемы и ограничения в процессе распознавания лиц, изучены этапы процесса распознавания лиц. В заключении отметим, что в зависимости от целей пользователя необходимо выбирать более удобный из методов, основываясь на входных данных и требуемого результатах.

Литература

1. *Bakhshi, Y.* A Study based on Various Face Recognition Algorithms / Y. Bakhshi, S. Kaur, P. Verma // International Journal of Computer Applications. – 2015. – Vol. 129, No. 13. – P. 16–20.

2. *Goyal, S.* A Review on Face Recognition Algorithms / S. Goyal, N. Batra // International Journal of Advanced and Innovative Research. – 2015. – Vol. 4, I. 12. – P. 150–153.

3. *Bhele, S. G.* A Review Paper on Face Recognition Techniques / S. G. Bhele, V. H. Mankar // International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). – 2012. – Vol. 1, I. 8. – P. 339–346.

4. *Лебеденко, Ю. И.* Биометрические системы безопасности / Ю. И. Лебеденко. – Москва : Директмедиа Паблишинг. – 2013. – 159 с.

5. Анализ существующих подходов к распознаванию лиц – URL: https://habr.com/ru/company/synesis/blog/238129/ (дата обращения 03.07.2020)

6. Распознавание лиц с использованием метода главных компонент и нейросети – URL: http://masters.donntu.org/2014/fknt/muradina/library/ (дата обращения 03.07.2020)

7. *Кузнецов, Д. А.* Классификация методов обнаружения и распознавания лица на изображении / Кузнецов Д. А., Никольский П. Г., Рачков Д. С., Кузнецов А. В., Хахамов А. П. // Научный результат. – 2019. – Т. 4, № 1. – С. 38–46.

Енокян Виктория Арамовна – студентка 4-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: vika-yenokyan@mail.ru

Леденева Татьяна Михайловна (научный руководитель) – д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета. Email: ledeneva-tm@yandex.ru

УДК 004.928

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОЙ АНИМАЦИИ ДЛЯ СОЗДАНИЯ МЕТОДИЧЕСКИХ МАТЕРИАЛОВ

С. А. Есина

Воронежский государственный университет

Введение

На сегодняшний день в учебном процессе невозможно обойтись без визуального представления информации. Использование анимации в обучении позволяет максимально заинтересовать учащихся и мотивировать их совершенствовать свои знания и навыки. С помощью технологий компьютерной анимации создаются приложения к электронным учебникам и презентациям. Эти методы особенно эффективны, так как внимание аудитории полностью фокусируется на анимированной картинке, что позволяет информации как можно более полно отложиться в памяти обучающихся.

В связи с тем, что создание анимации – достаточно долгий и ресурсоемкий процесс, на данный момент активно развиваются методы компьютерной анимации, что позволяет значительно ускорить и упростить создание анимационного продукта.

1. Анимация

1.1. Определение анимации

Анимация – технический прием создания иллюзии движения с помощью последовательности неподвижных изображений, или кадров[1]. В отличие от кинематографа, где объектами выступают, как правило, живые актеры, в мультипликации происходит работа только с неодушевленными предметами.

1.2. Виды и способы анимации

Рисованная, или двухмерная, анимация. Такие мультфильмы раньше рисовались с помощью специального стола с подсветкой и бумаги с калькой, а каждый кадр прорисовывался аниматором вручную. На сегодняшний день вместо таких столов используются компьютеры и специальное программное обеспечение.

Кукольная, или перекладная, или пластилиновая анимация. Такая анимация создается с помощью пластилиновых объектов, которые модифицируются между кадрами.

Motion capture (захват движения). Чаще всего этот вид анимации используется при создании фильмов или для анимации игровых персонажей. При создании анимации в этой технике на актера накладываются специальные световые датчики, а камеры вокруг актера фиксируют положение этих датчиков в пространстве. Далее с помощью компьютера происходит обработка положений этих точек, очищение их аниматором от паразитивных движений и дальнейшая передача на персонажа. Современные технологии позволяют в режиме реального времени увидеть, как движения актера будут выглядеть непосредственно на персонаже.

Трехмерная анимация. В отличие от двухмерной анимации, где каждый кадр приходится перерисовывать, аниматор управляет трехмерным персонажем с помощью анимационных кривых. Помимо самого аниматора, в создании такой анимации принимают участие многие

другие специалисты, которые занимаются моделированием персонажей, настройкой освещения и камер, созданием текстур и прочие[2].

1.3. Двенадцать принципов анимации

Все вышеперечисленные виды анимации различаются по технологиям создания, но все подчиняются основным двенадцати принципам анимации, предложенным на студии Disney аниматорами Олли Джонсоном и Френком Томпсоном[3].

Сжатие и растяжение. Все предметы в процессе движения сжимаются и растягиваются, сохраняя при этом свой объем.

Подготовка, или упреждение. Перед любым резким движением должна быть подготовка к нему. Например, перед прыжком в длину персонаж немного отходит назад. Подготовительное движение всегда совершается в направлении, противоположном задуманному[4].

Сценичность. Этот принцип осуществляется, когда движение хорошо читаемо и узнаваемо зрителем. Ни одна важная деталь не должна быть скрыта от внимания наблюдателя вследствие выбора неверного угла обзора. В достижении этого аниматору помогает работа с силуэтом персонажа.

Компоновка и планирование от позы к позе. До принятия этого принципа анимации аниматоры работали с позами последовательно, рисуя одну за другой. Принцип компоновки подразумевает создание аниматором ключевых поз и затем их уточнение в промежуточных кадрах.

Сквозное движение и захлест. Одно движение не должно полностью прекращаться до того момента, как начнется следующее. Здесь работает закон сохранения импульса: есть некоторая ведущая часть, и за ней следует другая, ведомая. В ведущей части возникает импульс, а затем передается ведомой и приводит ее в движение следом за ведущей. Захлест же означает пересечение движений, когда одно действие постепенно переходит в другое.

Плавное начало и завершение движения. Начало и конец действия должны наиболее выделяться относительно всего движения, поэтому аниматор делает на них акцент, выделяет на фоне остального.

Движение по аркам, или по дуге. Аниматор избегает механических движений вверх-вниз, так как в природе они практически невозможны вследствие пересечения различных движущих сил. Чем медленнее движение, тем глубже арка, чем быстрее – тем она более плоская. Даже резкое действие вроде падения будет начинаться с дуги, которая затем перетечет в прямую линию.

Дополнительное, или уточняющее, действие. Вторичные детали помогают в процессе действия раскрыть персонажа или сцену. Такие детали могут быть легко удалены без потери общего смысла, но они придают анимации завершенность.

Тайминг. Это расчет времени, за которое выполняется то или иное действие. Тайминг влияет на внимание зрителя, а также помогает передать размер и вес объекта.

Преувеличение и утрирование. С помощью утрирования аниматор подчеркивает главные движения. Находятся максимально выразительные черты, выделяется суть происходящего, чтобы зрителю было предельно ясно, что именно в сцене является основным действием.

Профессиональный рисунок. По одному взгляду на сцену зритель должен чувствовать вес, объем и равновесие. Аниматор в своей работе избегает симметрии, когда одна половина персонажа полностью копирует другую, так как подобная анимация выглядит механической.

Привлекательность. Персонаж должен постоянно приковывать и удерживать взгляд зрителя, даже если он не является основным. Вся сцена в целом также должна быть привлекательной и фокусировать на себе внимание[5].

2. Основы компьютерной анимации

2.1. Компьютерная анимация

Компьютерная анимация – вид анимации, которая создается при помощи компьютера. Так как компьютерная анимация является производной от компьютерной графики, она имеет те же способы создания: векторная, растровая, фрактальная, трехмерная.

2.2. Autodesk Maya 2019

Autodesk Maya – программный продукт для создания трехмерной анимации, моделирования и визуализации. Maya является стандартом программы для работы аниматора и обладает следующими достоинствами:

- Включает широкий набор инструментов для специалистов по анимации.
- Практически нет необходимости в подключении дополнительных инструментов, так как самые необходимые уже интегрированы разработчиками.
- Благодаря разнообразию инструментов, подходит для создания анимации разных уровней сложности.
- Maya позволяет начать работу без требования специальных навыков, в том числе программирования.

Интерфейс Maya 2019

Интерфейс программы является своего рода конструктором, и для удобства пользователя реализована возможность создавать свои собственные полки и добавлять на них инструменты, которые не находятся в быстром доступе, но часто требуются пользователю[6]. В рабочей области программы располагаются окна проекций. По умолчанию их четыре: вид сверху, вид спереди, вид сбоку и окно перспективы. Программа позволяет работать как в четырех видах одновременно, так и в каждом по отдельности. Пользователь может также создавать и настраивать свою камеру и работать только с ней. Это удобно, так как в большинстве случаев анимация делается именно под определенную камеру. Камера – это набор параметров, задающих в трехмерном пространстве положение наблюдателя, экрана и способ проецирования. В конкретной камере сцена может смотреться наиболее выгодно, в то время как в других видах упускаются важные детали, и анимация выглядит непривлекательно, что нарушает один из ее принципов. На рис. 1 показан фрагмент интерфейса программы, содержащий инструменты для аниматоров.

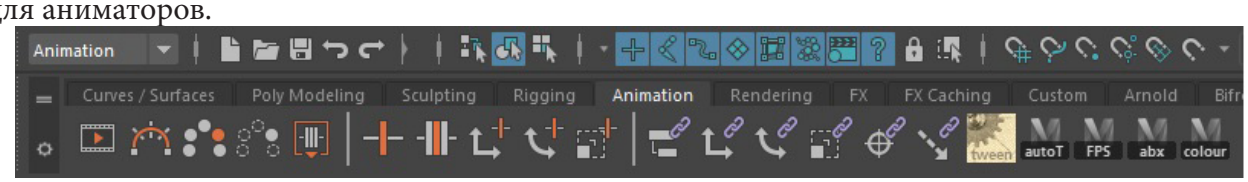


Рис. 1. Фрагмент интерфейса Maya 2019

Для создания анимации необходимо каждому объекту задать свои анимационные кривые, на основании которых они затем будут перемещаться.

Положение объекта в пространстве в определенный момент времени фиксируется с помощью анимационного ключа. Все ключи в Maya наносятся на специальную шкалу, на которой обозначены номера кадров. Эта шкала называется таймлайном. На рис. 2 находится пример расположения ключей на таймлайне.

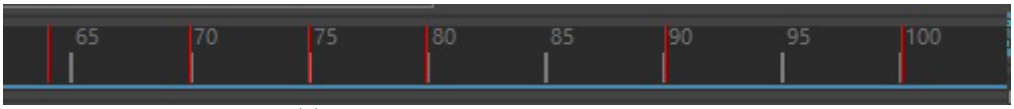


Рис. 2. Таймлайн с анимационными ключами

Анимационные ключи подвергаются различным видам редактирования: создание, удаление, перемещение, копирование и вставка. Несколько ключей также можно масштабировать, то есть увеличивать или уменьшать расстояние между ними.

Анимационная кривая – график зависимости какого-либо параметра объекта от времени. В зависимости от настроек Maya, анимационные кривые могут иметь различное представление, в том числе линейное или ступенчатое. Однако в анимации используются анимационные кривые в форме сплайнов [7]. Эти сплайны создаются программой с помощью автоматической интерполяции, где узлами выступают заданные аниматором анимационные ключи. С помощью перемещений, удалений и добавлений ключей аниматор редактирует анимационные кривые и тем самым создает анимацию.

Graph Editor – инструмент для работы с анимационными кривыми. Он позволяет аниматору более наглядно представлять все имеющиеся на данный момент анимационные кривые, в том числе относящиеся к разным параметрам объекта. С помощью Graph Editor можно редактировать анимационные ключи, положение объекта в пространстве и время, в которое объект оказался в определенном месте. На рис. 3 представлен интерфейс Graph Editor с расположенными в рабочей области анимационными кривыми.

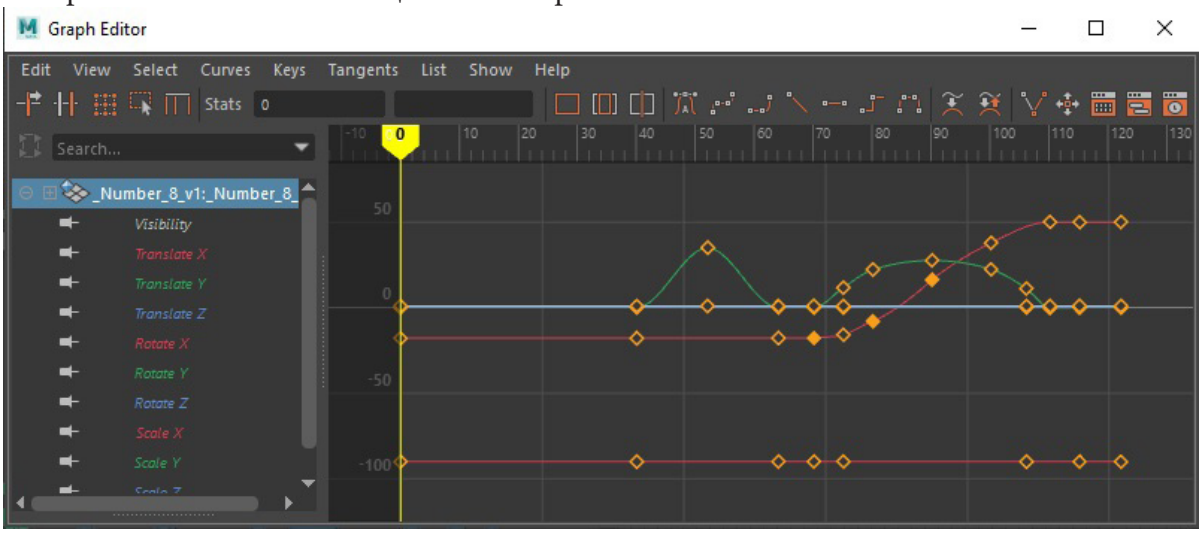


Рис. 3. Graph Editor

В левой части Graph Editor отображены все каналы, на которых существуют анимационные кривые. В правой части расположены непосредственно кривые с указанными на них ключами. В верхней части можно увидеть все возможные инструменты для работы с анимационными кривыми в Graph Editor

Тангенты – касательные, которые находятся в каждом анимационном ключе. С их помощью можно изменять вид анимационной кривой вручную. Такой способ доступен, если в настройках указано, что тангенты имеют вес. Тангенты в Graph Editor расположены справа и слева от каждого анимационного ключа. Выделив любой из тангентов, можно изменять его положение относительно ключа или растягивать его, тем самым меняя форму кривой. На рис. 4 показан пример тангентов.

По умолчанию два тангента связаны друг с другом, но с помощью клавиши Break tangents можно разделить их и работать с каждым в отдельности независимо друг от друга. Чтобы снова объединить тангенты, используется клавиша Unify tangents. На практике тангенты практически не используются, так как в процессе производства анимационного продукта одна и та же сцена может оказаться в руках нескольких специалистов анимации, что делает затруднительным анализ каждого анимационного ключа и выходящих из него тангентов. Обычно работа ведется только с положением анимационных ключей в плоскости рабочей области Graph Editor.

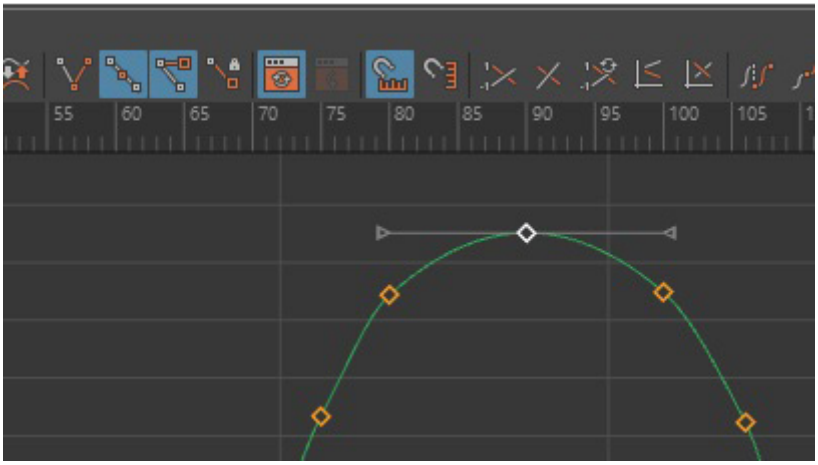


Рис. 4. Тангенты

3. Этапы создания компьютерной анимации

3.1. Выбор сценария. Подготовка сцены

Для создания анимации необходимо каждому объекту задать свои анимационные кривые, на основании которых они затем будут перемещаться. Для этого вначале создается сценарий, а затем происходит процесс создания анимации.

В рамках данного проекта в качестве примера была выбрана быстрая сортировка. Итоговая анимация будет визуализировать сортировку массива, содержащего цифры от 0 до 9, расположенных в произвольном порядке. Результатом выполнения алгоритма будет массив из упорядоченных по возрастанию чисел. Анимация будет визуализировать каждый шаг применения данного алгоритма к конкретному массиву. На рис. 5 представлена начальная сцена в рабочей среде Maya.

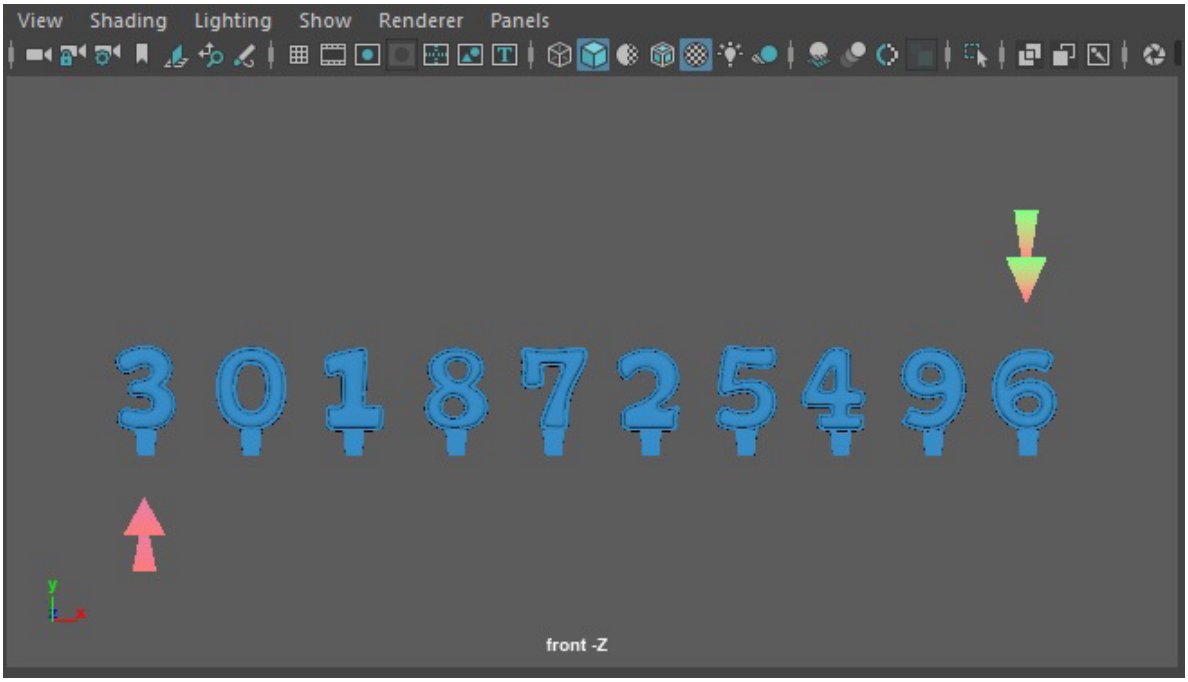


Рис. 5. Начальная сцена анимации

3.2. Реализация сценария

В соответствии с заданным алгоритмом объекты сцены, характеризующие определенные элементы массива, в конкретный момент времени будут принимать какое-либо положение в пространстве. Это означает, что объект меняет свои параметры перемещения, за которые отвечают каналы Translate X, Translate Y и Translate Z. Кроме того, за положение в пространстве отвечают параметры вращения объектов, которые задают степень вращения вокруг одной из координатных осей. За эти параметры отвечают значения каналов Rotate X, Rotate Y и Rotate Z. Также в соответствии с принципом сжатия и растяжения объекты в процессе перемещения могут менять и параметры масштаба, за которые отвечают каналы Scale X, Scale Y и Scale Z. Значения параметров объекта расположены в правой рабочей области и могут как задаваться вручную, так и рассчитываться программой в случае промежуточных кадров между заданными ключами. Наличие ключа в конкретном кадре характеризуется ярко-красной меткой рядом с определенным параметром. Бледно-красная метка означает, что значение параметра было изменено, но не зафиксировано созданием анимационного ключа. Обычно это касается промежуточных кадров, либо случаев, когда аниматор забыл поставить ключ. На рис. 6 продемонстрирован пример значений параметров объекта в одном из ключевых кадров сцены, где ключи стоят на каналах Rotate Y и Scale Z.

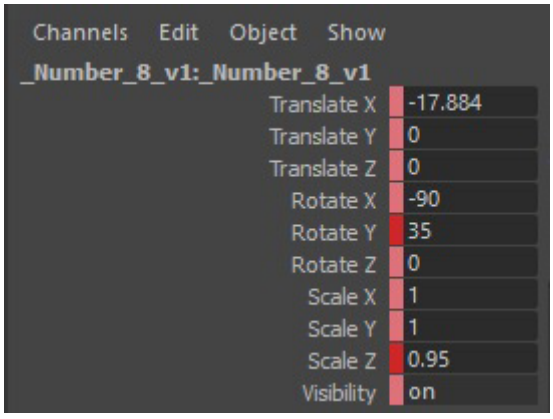


Рис. 6. Значения параметров объекта

Для остальных каналов значения параметров будут промежуточными, а значит, они рассчитаны программой автоматически, исходя из значений в двух соседних кадрах. С помощью изменений параметров разных объектов в конкретные моменты времени и создается сцена, наглядно демонстрирующая процесс быстрой сортировки на заданном массиве. После расстановки ключевых кадров наступает процесс редактирования полученных анимационных кривых с помощью инструмента Graph Editor. Согласно одному из принципов анимации, движению по дуге, все анимационные кривые, отвечающие за положение объекта в пространстве, должны представлять из себя арки. Также в процессе редактирования кривых важно помнить и о других принципах анимации, например, плавное начало и окончание движения. Работа с анимационными кривыми позволяет демонстрировать выполнение принципов анимации на практике как можно более наглядно.

3.3. Рендеринг

Когда сценарий полностью реализован, а техничность его выполнения соответствует стандартам и принципам анимации, наступает процесс рендеринга – визуализации сцены и перехода от трехмерной графики к двухмерной. В Maya за это отвечает инструмент Playblast.

Перед процессом рендеринга необходимо выставить настройки итогового изображения, которые подходят под заданную задачу. В рамках этой работы использовано качество видео HD 720. Частота кадров задана согласно распространенному кинематографическому стандарту, 24 кадра в секунду (fps). Необходимо выбрать окно проекции, с которой будет происходить рендеринг. В этом окне может находиться как один из видов, так и настраиваемая аниматором камера. Инструмент Playblast можно найти, щелкнув правой кнопкой мыши по таймлайну. Там же находятся необходимые настройки. Учитывая, что настройки самого изображения были заданы заранее, в графе Display size необходимо указать From Render Settings. Здесь же можно задать путь, по которому сохранится итоговый файл. В случае отсутствия пути программа сохранит результат в папке по умолчанию, которая задана в ее настройках. Результатом проделанной работы будет видео, на котором изображен процесс быстрой сортировки.

Заключение

Данный проект в дальнейшем может быть использован в образовательном процессе для ознакомления обучающихся с темой «быстрая сортировка». При выполнении проекта были изучены основные принципы анимации, а также методы создания анимации с помощью компьютера и специализированного программного обеспечения, в частности программы Autodesk Maya 2019.

Так как цифровые методы активно внедряются в современный процесс обучения, данный проект актуален и будет полезен при разработке образовательных программ.

Литература

1. *Уильямс, Р.* Аниматор: набор для выживания / Р. Уильямс. – Москва : Эксмо, 2020. – 392 с.

2. Wizart School – Школа анимации. – URL: https://wizartschool.com/ (дата обращения: 28.03.2021)

3. *Томас, Ф.* The Illusion of Life: Disney Animation / Ф. Томас, О. Джонстон. – Disney Editions : Rev Sub edition, 1995. – 548 с.

4. *Блэр, П.* Cartoon Animation / П. Блэр. – Мишен-Вьехо : Walter Foster Publishing, 1994. – 224 с.

5. *Хитрук, Ф. С.* Профессия – аниматор / Ф. С. Хитрук. (В 2 тт., т. 2.) – Москва : Гаятри, 2007. – 304 с.

6. Autodesk Knowledge Network. – URL: https://knowledge.autodesk.com/ru/ (дата обращения: 18.03.2021)

7. eduCBA. – URL: https://www.educba.com/ (дата обращения: 10.04.2021)

Есина Светлана Александровна – студентка 4-го курса кафедры Математического обеспечения ЭВМ Воронежского государственного университета. E-mail: lana@aport.ru

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры Математического обеспечения ЭВМ Воронежского государственного университета. E-mail: bolotova.svetlana@gmail.com

УДК 004.04

ИССЛЕДОВАНИЕ МОДЕЛИ МЕХАНИЗМА ПЕРЕДАЧИ И СЕГМЕНТИРОВАНИЯ ВИДЕОПОТОКА С IP-КАМЕР

Н. Е. Жарковский

Воронежский государственный университет

Введение

IP-камеры широко применяются для решения задач системы видеонаблюдения. Особенностью таких камер является передача видеопотока в цифровом формате по сети Ethernet. Задача заключается в просмотре видео, передаваемого камерой, как в режиме реального времени, так и за заданный промежуток времени.

Решением данной задачи является создание медиасервера, через который будет проходить весь видеопоток. Сервер должен уметь сохранять видео у себя в хранилище и ретранслировать его клиенту в режиме реального времени. В качестве графического интерфейса пользователя будет выступать Web-приложение.

В статье описывается решение данной задачи с использованием протокола RTSP и динамической адаптивной потоковой передачи через HTTP.

1. Механизм передачи видеопотока с камеры

Почти все IP-камеры умеют передавать видео при помощи RTSP (англ. real time streaming protocol, то есть потоковый протокол реального времени). Медиасервер должен сначала получить поток RTSP, затем преобразовать его в поток, который может воспроизводиться браузерами. Это может быть HTTP Live Streaming (HLS) или динамическая адаптивная потоковая передача через HTTP (MPEG-DASH). Данные потоки могут воспроизводиться в браузере при помощи HTML5 video, если поддерживается Media Source Extensions (MSE).

Протоколы HLS и MPEG-DASH примечательны тем, что поток может быть разбит на сегменты. В случае HLS это m3u8-плейлисты и сегменты в формате MPEG-TS (*.ts). В MPEG-DASH вместо плейлистов – MDP-манифест в XML формате и сегменты *.m4s.

Для преобразования RTSP потока в MPEG-DASH можно воспользоваться Ffmpeg – набор свободных библиотек с открытым исходным кодом, которые позволяют записывать, конвертировать и передавать цифровые аудио- и видеозаписи в различных форматах. В разных языках программирования можно воспользоваться уже существующими обертками над FFmpeg или реализовать свою. Также доступен вариант использования уже готовой сборки FFmpeg.exe.

2. Команда для передачи и сегментирования видеопотока

2.1. Пример преобразования RTSP в MPEG-DASH

В качестве примера возьмем видеоролик, который находится в открытом доступе по адресу: rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny_115k.

Для преобразования RTSP потока в MPEG-DASH при помощи Ffmpeg необходимо воспользоваться командой (Листинг 1).

Листинг 1

```
ffmpeg
-i rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny_115k.mov
-f dash
-seg_duration 2000
-use_template 1
-use_timeline 0
-init_seg_name init-stream$RepresentationID$.$ext$
-media_seg_name chunk-stream$RepresentationID$-$Number%05d$.$ext$
manifest.mpd
```

2.2. Параметры команды

Описание параметров команды, которая использовалась для преобразования RTSP в MPEG-DASH:

1) i <input_video> – видео, которое необходимо конвертировать;
2) f dash – формат преобразования медиапотока;
3) min_seg_duration 2000 – установка продолжительности сегментов;
4) use_template 1 – разрешение шаблонного имени для сегментов;
5) use_timeline 0 – запрет на использование SegmentTimeline в имени сегмента;
6) init_seg_name – шаблон имени сегмента инициализации;
7) media_seg_name – шаблон имени сегмента;
8) manifest.mpd – имя плейлиста.

2.3. Результат выполнения команды

После выполнения команды (Листинг 1) формируются следующий набор файлов:

1. manifest.mpd – media presentation description. Представляет собой XML-файл, в котором указаны ссылки на различные качества медиа-контента с помощью HTTP URL (Uniform Resource Locators). Эта структура обеспечивает привязку сегментов к битрейту, разрешению, времени начала, длительности.

2. init-stream0.m4s и init-stream1.m4s – сегменты для инициализации (0 – видео, 1 – аудио).

3. chunk-stream0-*.m4s и chunk-stream1-*.m4s – медиасегменты для видео и аудио потоков соответственно. Размер каждого сегмента 2 секунды.

В результате, каждый клиент сначала запрашивает MPD, тем самым получая временную и структурную информацию о контенте, и на основе этой информации запрашивает отдельные сегменты для воспроизведения, которые в большей степени соответствуют возможностям и требованиям.

Полученные фрагменты можно сохранять на файловой системе, делая временную пометку для файла manifest.mpd.

Заключение

Чтобы организовать видеонаблюдение с возможностью сегментирования и сохранения видео, необходимо использовать MPEG-DASH. В результате RTSP поток будет преобразован и разбит на небольшие фрагменты продолжительностью 2 секунды. Информация о сегментах и плейлистах может храниться на файловой системе или в базе данных. Это позволит организовать потоковую передачу, как в реальном времени, так и за определенный временной промежуток с возможностью перемотки и синхронизации нескольких видеопотоков.

Литература

1. *Красилов, А. Н.* Исследование передачи коротких видеопотоков mpeg-dash в сетях wi-fi / А. Н. Красилов, М. В. Любогощев // ИТиС-2016. – Санкт-Петербург, 2016. – С. 679-686.

2. Dynamic adaptive streaming over HTTP (DASH) – Part 5: Server and network assisted DASH (SAND). Standard: ISO/IEC 23009-5:2017. – Vernier, Geneva, Switzerland, 2017.

3. Romero L. R. A dynamic adaptive http steaming video service for google android / L. R. Romero // Royal Institute of Technology (KTH). – Stockholm, 2011.

4. *Spiteri, R. U. K.* Bola: Near-Optimal Bitrate Adaptation For Online Videos / R. U. K. Spiteri, R. K. Sitaraman // IEEE INFOCOM 2016. – pp. 1-9.

5. *Stockhammer, T.* Dynamic adaptive streaming over HTTP: standards and design principles / T. Stockhammer // MMSys '11: Proceedings of the second annual ACM conference on Multimedia systems. – 2011. – pp. 133-144.

6. FFmpeg. A complete, cross-platform solution to record, convert and stream audio and video. – Режим доступа: https://www.ffmpeg.org. – (Дата обращения: 01.04.2021)

Жарковский Никита Евгеньевич – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: nikita.zharkovsky@gmail.com

Горбенко Олег Данилович (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: oleg_dan@mail.ru

УДК 621.391.6

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ RFID ДЛЯ ОПРЕДЕЛЕНИЯ ПОЛОЖЕНИЯ ОБЪЕКТОВ В РАМКАХ ЗДАНИЯ

А. В. Золотарев

Воронежский государственный университет

Введение

Технологии RFID (Radio Frequency Identification), или РЧИ (Радиочастотной Идентификации), позволяют дистанционно идентифицировать любой физический объект с помощью, прикрепленной к нему радиочастотной метки. Метка содержит микрочип, хранящий информацию об объекте (серийный номер, модель, предприятие производитель, стоимость, вес, дату изготовления и т. д.) и антенну, передающую эту информацию на RFID-считыватели.

Метки, содержащие только микрочип и антенну, называются пассивными, так как они используют электромагнитную энергию, получаемую от считывателя для питания микрочипа. Также существуют метки, содержащие автономные элементы питания (батареи). Активные метки используют батарею для постоянной самостоятельной передачи сигнала. Полупассивные метки используют электромагнитную энергию считывателя для пробуждения, и энергию батареи – для передачи сигнала.

RFID – способ автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются данные, хранящиеся в так называемых транспондерах, или RFID-метках.

1. Radio Frequency Identification

Любая RFID-система состоит из считывающего устройства (считыватель, ридер или интеррогатор) и транспондера (он же RFID-метка, иногда также применяется термин RFID-тег).По дальности считывания RFID-системы можно подразделить на системы:

- ближней идентификации (считывание производится на расстоянии до 20 см);
- идентификации средней дальности (от 20 см до 5 м);
- дальней идентификации (от 5 м до 300 м)

Большинство RFID-меток состоит из двух частей. Первая – интегральная схема (ИС) для хранения и обработки информации, модулирования и демодулирования радиочастотного (RF) сигнала и некоторых других функций. Вторая – антенна для приёма и передачи сигнала.

С введением RFID-меток в повседневную жизнь связан ряд проблем. Например, потребители, не обладающие считывателями, не всегда могут обнаружить метки, прикреплённые к товару на этапе производства и упаковки, и избавиться от них. Хотя при продаже, как правило, такие метки уничтожаются, сам факт их наличия вызывает опасения у правозащитных[1] и религиозных[2] организаций.

Уже известные приложения RFID (бесконтактные карты в системах контроля и управления доступом, системах дальней идентификации и в платёжных системах) получают дополнительную популярность с развитием интернет-услуг.

2. Дальняя RFID идентификация

Дальняя идентификация – технология радиочастотной идентификации (RFID), которая позволяет регистрировать снабженные активными радиочастотными метками объекты на больших расстояниях (до 50 метров) от регистрирующего (считывающего) устройства.

Использование технологии позволяет решить большое количество задач в различных областях деятельности: идентификация автотранспорта, системы безопасности, логистика, складской учёт, автоматизация производств и т. д.

Системы, используемые для дальней идентификации, как правило состоят из трех элементов:

- Считыватель
- Радиочастотные метки
- Программное обеспечение

На объект, который необходимо контролировать, устанавливается радиочастотная метка. При попадании объекта с меткой в зону действия считывателя, считыватель принимает от метки содержащуюся в ней информацию об объекте и передаёт её в вычислительное устройство, снабженное специальным программным обеспечением.

Для радиочастотной идентификации могут применяться два типа меток, в зависимости от наличия в них автономного батарейного питания: активные и пассивные. Применение в системах дальней идентификации активных меток, работающих на сверхвысоких частотах и автономном батарейном питании отличает эти системы от стандартных пассивных RFID-систем и делает возможным идентификацию объектов на больших расстояниях (до 50 м).

Существует два основных режима RFID дальней идентификации:

- *Режим доступа*: в этом режиме считыватель интегрируется в стандартную систему контроля и управления доступом (СКУД). Как только в зоне его действия оказывается метка, находящаяся у человека или установленная в автомобиле, информация с неё через считыватель поступает в контроллер СКУД, который открывает преграждающее устройство. Система дальней идентификации не требует остановки машины, открытия окон и близкого поднесения метки к считывателю, что делает её максимально комфортной для контроля доступа на платные парковки, автостоянки и для пропуска VIP-персон.
- *Режим мониторинга*: в режиме мониторинга считыватели в комплекте с метками позволяют решать задачи по учёту наличия и перемещения контролируемых объектов на заданной территории в реальном времени. При этом возможна запись, хранение получаемой информации и генерирование тревожных сигналов при появлении или исчезании метки или при исчезании с контролируемой территории.

Режим мониторинга используется при решении следующих задач:

- Отслеживание движения транспорта
- Определение местоположения людей
- Контроль личного состава в районе заданной области (КПП, пост охраны и т. п.). Регистрируется вход/выход носителя метки в регистрируемую область. Возможность генерирования сигнала тревоги в случае покидания поста
- Контроль выноса дорогостоящего оборудования. При несанкционированном выносе оборудования из охраняемого помещения генерируется сигнал тревоги
- Контроль спортивного инвентаря (мотоциклы, картинги, велосипеды и т. п.). При несанкционированном изъятии инвентаря из охраняемой области возникает сигнал тревоги
- Контроль и учёт рабочего времени служащих. Регистрируются все временные моменты, связанные с появлением и исчезновением с территории офиса служащих

Заключение

В заключение можно отметить, что технология RFID уже достаточно распространена для определения относительного местоположения сотрудников и контроля их доступа в различные зоны предприятия. Стоит отметить, что наиболее распространена ближняя RFID идентификация персонала, однако на специфичных производствах, а также на охраняемых объектах, таких как автомобильные парковки используется дальняя RFID идентификация как обладающая преимуществами, о которых шла речь в статье.

Литература

1. *Власов М.* RFID: 1 технология – 1000 решений: Практические примеры использования RFID в различных областях. – М. : Альпина Паблишер, 2014. – 218 с.

2. *Лахири С.* RFID. Руководство по внедрению = The RFID Sourcebook / С. Дудников. – М. : Кудиц-Пресс, 2007. – 312 с.

3. *Бхуптани М., Морадпур Ш.* RFID-технологии на службе вашего бизнеса = RFID Field Guide: Deploying Radio Frequency Identification Systems / Н. Троицкий. – М. : «Альпина Паблишер», 2007. – 290 с.

4. *Финкенцеллер К.* Справочник по RFID. – М. : Издательский дом «Додэка-XXI», 2008. – 496 с.

Золотарев Александр Витальевич – студент 2-го курса магистратуры кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: fuwaaika@ inbox.ru

Сафронов Виталий Владимирович (научный руководитель) – канд. техн. наук, доц., доцент кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: vitolik@bk.ru

УДК 515.14, 004.8

АППРОКСИМАЦИЯ ФУНКЦИЙ С ПОМОЩЬЮ НЕЧЕТКИХ СИСТЕМ

А. Е. Золотухин

Воронежский государственный университет

Введение

Нечеткие системы применяются для решения многих прикладных задач, в том числе и задач управления. Теоремы о возможности аппроксимации непрерывных функций при помощи нечетких систем, сделали методы, использующие их, математически обоснованными.

Основополагающие результаты по теории аппроксимации функций были получены в девятнадцатом веке К. Вейерштрассом и П. Л. Чебышевым. В двадцатом веке в теорию приближений внесли академик А. Н. Колмогоров и С. Н. Бернштейн. При создании этой теории в качестве аппроксимирующих агрегатов использовались алгебраические многочлены и тригонометрические полиномы, для аппроксимации непериодических и периодических функций соответственно. В дальнейшем развитие вычислительной техники позволило использовать для приближенных вычислений другие методы, основанные на использовании искусственных нейронных сетей и нечетких систем.

Первые теоремы о возможности использования для аппроксимации непрерывных функций однонаправленных нейронных сетей с одним скрытым слоем получены в работах [1, 2]. Первые теоремы о возможности использования для аппроксимации функций нечетких систем доказаны в работах [3, 4].

В данной работе приводятся некоторые результаты, относящиеся к приближению функций одной или нескольких действительных переменных алгебраическими многочленами, и носят вспомогательный характер. В работе дается обзор результатов по приближению функций нечеткими системами.

1. Аппроксимация функций алгебраическими многочленами

Пусть $C[0,1]$ – пространство непрерывных функций $f:[0,1]\to\mathbb{R}$. Норма функции $f\in C[0,1]$ определяется соотношением

$$\|f\|_\infty = \sup_{x\in[0,1]} |f(x)|.$$

Через Π_n обозначим пространство алгебраических многочленов степени не выше n. Наилучшем приближением функции $f\in C[0,1]$ алгебраическим многочленом степени не выше n называется величина

$$E_n(f) = \inf_{P_n\in\Pi_n} \|f-P_n\|_\infty .$$

Следующая теорема принадлежит Вейерштрассу

Теорема 1. *Для любой функции* $f\in C[0,1]$ $E_n(f)\to 0$ *при* $n\to\infty$.

Доказательство теоремы Вейерштрасса можно найти в книге [5].

Модуль непрерывности функции $g\in C[0,1]$ определяется следующим образом. При $\delta>0$

$$\omega(g,\delta) = \sup_{\substack{x,y\in[0,1]\\ |x-y|\le\delta}} |g(x)-g(y)|.$$

В терминах модулей непрерывности функции f и ее производных (если они существуют) можно узнать, насколько быстро последовательность $E_n(f)$ стремится к нулю в зависимости от гладкости функции f.

Теоремы 2 и 3 принадлежат Джексону

Теорема 2. *Если функция* $f \in C[0,1]$, *то при любом натуральном* n

$$E_n(f) \le C_0 \omega\left(f, \frac{1}{n}\right),$$

где C_0 *– некоторая абсолютная константа.*

Теорема 3. *Если функция* $f \in C[0,1]$, *имеет r непрерывных производных на отрезке* $[0,1]$, *то при любом* $n > r$

$$E_n(f) \le \frac{C_r}{n^r} \omega\left(f^{(r)}, \frac{1}{n-r}\right),$$

где константа C_r *зависит только от* r.

Доказательства теорем 2 и 3 приведены в книге [5]. Теоремы 2 и 3 относятся к прямым теоремам теории приближений.

Пусть функция $f \in C[0,1]$ имеет r непрерывных производных на отрезке $[0,1]$. Существует ли такой алгебраический многочлен P_n, приближающий функцию f, производные которого $P_n', \ldots, P_n^{(r)}$ приближают производные $f_n', \ldots, f_n^{(r)}$ соответственно? Данная задача называется задачей о совместном приближении алгебраическими многочленами функции и ее производных.

Для функции $f \in C[0,1]$ С. Н. Бернштейном был введен многочлен

$$B_n(f;x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) C_n^k x^k (1-x)^{n-k}.$$

Доказательство того, что многочлены Бернштейна дают решение задачи о совместном приближении, можно найти в книге [6]. Но скорость приближения функции многочленами Бернштейна значительно медленнее, чем та, которая дается теоремой [7].

Оператор $L_n : C[0,1] \to C[0,1]$ называется линейным, если для любых функций $f, g \in C[0,1]$ и любых $a, b \in \mathbb{R}$

$$L_n(af + bg) = aL_n(f) + bL_n(g).$$

Теорема 4 принадлежит П. П. Коровкину [14].

Теорема 4. *Пусть* $\{L_n\}$ *– последовательность линейных операторов из* $C[0,1]$ *в* $C[0,1]$. *Рассмотрим функцию* $g_k(x) = x^k$. *Тогда если*

$$\|g_k - L_n(g_k)\|_\infty \to 0 \ \text{при} \ n \to \infty, \ k = 0,1,2,$$

то для любой функции $f \in C[0,1]$

$$\|f - L_n(f)\|_\infty \to 0 \ \text{при} \ n \to \infty.$$

Задача приближения функций алгебраическими многочленами изучена не только для равномерной метрики, но и для интегральных метрик.

Пространство $L_p[0,1]$, $1 \le p < \infty$, состоит из измеримых по Лебегу функций $f : [0,1] \to \mathbb{R}$, для которых

$$\int_0^1 |f(x)|^p \, dx < \infty.$$

Функции, различающиеся лишь на множестве меры ноль, отождествляются. Норма функции $f \in L^p[0,1]$ определяется следующим образом:

$$\|f\|_p = \left(\int_0^1 |f(x)|^p dx \right)^{1/p}.$$

Для наилучшего приближения $f \in L^p[0,1]$ алгебраическими многочленами в метрике L^p также имеются теоремы типа Джексона [8].

Значительная часть результатов о приближении функций одного действительного переменного в той или иной степени обобщена для функций нескольких действительных переменных [18].

Рассмотрим компактное множество $K \subseteq \mathbb{R}^d$. Пусть $C(K)$ – пространство непрерывных функций $f : K \to \mathbb{R}$ с равномерной нормой

$$\|f\|_\infty = \sup_{x \in K} |f(x)|.$$

Пусть M – линейное пространство пространства $C(K)$, и выполняются следующие условия:

1. Подпространство M является алгеброй.

2. Для любых несовпадающих точек $x, y \in K$, найдется функция $f \in M$ такая, что $f(x) \neq f(y)$.

3. Для любой точки $x \in K$ найдется функция $f \in M$ такая, что $f(x) \neq 0$.

Теорема Стоуна.

Теорема 5. *Подпространство M является плотным в пространстве $C(K)$ относительно равномерной метрики.*

Доказательство теоремы можно найти в книге [9].

Отметим, что пространство всех алгебраических многочленов относительно $x_1, ..., x_d$ удовлетворяет условиям 1–3. Поэтому верен многомерный аналог теоремы 1: любую функцию из пространства $C(K)$ можно сколь угодно точно приблизить алгебраическими многочленами в равномерной метрике.

По направлениям, обозначенным в данном пункте, развиваются и теория приближения искусственными нейронными сетями, и теория приближения нечеткими системами. Преимущества данных подходов заключается в большей однородности аппроксимирующих агрегатов, чем та, которая есть у одночленов $x_1^{k_1}, ..., x_1^{k_d}$ при различных $k_1, ..., k_d$, а также в меньшей зависимости от конкретной функциональной формы.

2. Аппроксимация функций нечеткими системами

Пусть A_j^l, $l = 1, ..., d$, $j = 1, ..., n$ – нечеткие множества, универсальным множеством является $\mathbb{R}$ (определение нечеткого множества в работе [10]). Через $\mu_{A_j^l}$ обозначим функцию принадлежности нечеткого множества A_j^l. Соответственно, $\mu_{A_j^l} : \mathbb{R} \to [0,1]$. Как и в предыдущем пункте рассматривается задача построения для некоторой функции аппроксимирующего агрегата из определенного класса. Этот агрегат должен каждому значению $x = (x_1, ..., x_d)$ из некоторого множества, принадлежащего $\mathbb{R}^d$, ставить в соответствие действительное число y.

Нечеткая система состоит из n нечетких правил вида:

$$R_j : ЕСЛИ\ (x_1 = A_j^1)\ И ... И\ (x_d = A_j^d)\ ТО\ y = a_j^0;\ j = 1, ..., n. \quad (1)$$

Здесь $a_j^0 \in \mathbb{R}$. Если консеквент имел бы вид $y = A_j^0$, где A_j^0 – некоторое нечеткое множество, то нечеткая система (1) называется системой Мамдини.

Пусть

$$\xi_j(x) = \frac{\prod_{l=1}^{d} \mu_{A_j^l}(x_l)}{\sum_{k=1}^{n} \prod_{l=1}^{d} \mu_{A_k^l}(x_l)}, j = 1, ..., n,$$

и

$$f_n(x) = \sum_{j=1}^{n} a_j^0 \xi_j(x). \quad (2)$$

Если будет доказано, что нечеткая система (1), (2) является универсальным аппроксиматором, то это будет обозначать, что и нечеткая система Мамдани будет являться универсальным аппроксиматором.

Пусть компактное множество $K \subseteq \mathbb{R}^d$. Без ограничения общности будем считать, что $K \subseteq [0,1]^d$. Выберем произвольную функцию $f \in C(K)$ и произвольное $\varepsilon > 0$.

Воспользовавшись теоремой 5, найдем алгебраический многочлен P, аппроксимирующий функцию f в равномерной метрике на K с точностью $\varepsilon/2$, т. е.

$$\sup_{x \in K} |f(x) - P(x)| < \frac{\varepsilon}{2}.$$

Пусть m – натуральное число, и каждое из целых чисел $q_1, ..., q_m$ может принимать значения $0, 1, ..., m$. Через Q_m обозначим множество точек $q = (q_1, ..., q_d)$. Таким образом, множество Q_m содержит $(m+1)^d$ точек. При $q \in Q_m$ через q/m обозначим точку $\left(\frac{q_1}{m}, ..., \frac{q_d}{m}\right)$, принадлежащую множеству $[0,1]^d$.

При каждом $q \in Q_m$ будем считать, что $A_q^1, ..., A_q^d$ – нечеткие множества. Функцию принадлежности нечеткого множества A_q^l обозначим $\mu_{A_j^l}$. Как и раньше, универсальным множеством является $\mathbb{R}$.

Нечеткая система состоит из $n = (m+1)^d$ нечетких правил.

$$R_q : ЕСЛИ\ (x_1 = A_q^1)\ И ... И\ (x_d = A_q^d)\ ТО\ y = P\left(\frac{q}{m}\right), \quad q \in Q_m.$$

Тогда

$$\xi_q(x) = \frac{\prod_{l=1}^{d} \mu_{A_q^l}(x_l)}{\sum_{q \in Q_m} \prod_{l=1}^{d} \mu_{A_q^l}(x_l)}, \ j = 1, ..., n, \tag{3}$$

$$f_n(x) = \sum_{q \in Q_m} P\left(\frac{q}{m}\right) \xi_j(x). \tag{4}$$

Теперь необходимо наложить ограничения на нечеткие множества A_q^l. Будем считать что знаменатель дроби (3) положителен при любом $x \in [0,1]^d$. При этом условии система функций $\xi_q(x)$ является разбиением единицы для $[0,1]^d$. Второе условие: существуют последовательности неотрицательных чисел $\alpha_m \to 0$ и $\beta_m \to 0$ при $m \to \infty$ такие, что при любом $x \in [0,1]^d$

$$\sum_{q \in M_2(x)} \xi_q(x) < \alpha_m,$$

где

$$M_1(x) = \left\{ q \in Q_m : \left\| x - \frac{q}{m} \right\| \le \beta_m \right\},$$

$$M_2(x) = \left\{ q \in Q_m : \left\| x - \frac{q}{m} \right\| > \beta_m \right\}.$$

Очевидно, что функции принадлежность $\mu_{A_j^l}$ можно построить так, чтобы эти два условия выполнялись, если в качестве класса функций принадлежности рассматривать класс всех кусочно-линейных функций.

Теорема 6. *Существует функция f_n вида (4) такая, что*

$$\sup_{x \in K} |f(x) - f_n(x)| < \varepsilon.$$

Доказательство теоремы 6. Существуют константы $C_1 > 0, C_2 > 0$ такие, что

$$\left|P(x)\right| < C_1, \left\|grad\,(Px)\right\| < C_2$$

При любом $x \in [0,1]^d$.

Зафиксировав натуральное число m и $x \in [0,1]^d$, положим

$$\gamma(x) = \sum_{q \in M_2(x)} \xi_q(x).$$

Тогда на основании выражения (4)

$$\left|P(x) - f_n(x)\right| = \left|P(x) - \sum_{q \in Q_m} P\left(\frac{q}{m}\right)\xi_j(x)\right| \le \left|P(x) - \sum_{q \in M_1(x)} P\left(\frac{q}{m}\right)\xi_j(x)\right| + \left|\sum_{q \in M_2(x)} P\left(\frac{q}{m}\right)\xi_j(x)\right|.$$

Второе слагаемое правой части неравенства не превосходит $C_1\alpha_m$. Поэтому

$$\left|P(x) - f_n(x)\right| \le \left|\gamma(x)P(x) + \sum_{q \in M_1(x)} \left(P(x) - P\left(\frac{q}{m}\right)\right)\xi_j(x)\right| + C_1\alpha_m \le$$

$$\le \sum_{q \in M_1(x)} \left|P(x) - P\left(\frac{q}{m}\right)\right|\xi_j(x) + 2C_1\alpha_m.$$

Воспользовавшись тем, что

$$\left|P(x) - P\left(\frac{q}{m}\right)\right| \le \left(\sup_{y \in [0,1]^d} \left\|gradP(y)\right\|\right)\left\|x - \frac{q}{m}\right\|,$$

получаем

$$\left|P(x) - f_n(x)\right| \le C_2\beta_m + 2C_1\alpha_m.$$

Откуда

$$\left\|P(x) - f_n\right\|_\infty \le C_2\beta_m + 2C_1\alpha_m.$$

При достаточно большом m

$$C_2\beta_m + 2C_1\alpha_m < \frac{\varepsilon}{2}.$$

Далее,

$$\sup_{x \in K}\left|f(x) - f_n(x)\right| \le \sup_{x \in K}\left|f(x) - P(x)\right| + \sup_{x \in K}\left|P(x) - f_n(x)\right| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Теорема 6 доказана.

В доказательстве теоремы 6 при больших значениях размерности пространства d число нечетких правил может быть очень большим. В данной ситуации наибольший интерес представляет снижение количества правил в нечеткой системе при сохранении точности аппроксимации.

Доказательство теоремы 6 дословно повторяется, если взять в качестве t-нормы не произведение, а минимум.

Заключение

Теория приближения функции – это большое и важное научное направление, появление которого связано с именами К. Вейерштрасса и П. Л. Чебышева. Результаты и методы данной теории находят применение в различных прикладных исследованиях.

Потребности практических задач привели к увеличению числа методов для аппроксимации функций, поскольку во многих задачах оказалось недостаточно алгебраических многочленов и тригонометрических полиномов.

При этом влияние классических разделов теории приближения функций очень сильно влияет на развитие новых методов. Доказываются те же теоремы, что и в классических разделах: об увеличении скорости сходимости, при повышении гладкости функции; о совместном приближении функции и ее производных и прочие. При этом метод аппроксимации функции при помощи нечетких систем теснее и все чаще применяется при решении практических задач.

Литература

1. *Cybenko G.* Approximation by superpositions of sigmoidal function // Mathematics of Control, Signals, and Systems. – 1989. – Vol. 2. – P. 303–314.

2. *Funahashi K. I.* On the approximate realization of continuous mappings by neural networks // Neural Networks. – 1989. – Vol. 2. – P. 183–192.

3. *Wang L.-X.* Fuzzy systems are universal approximators // in: Proc. IEEE 1992 Intern. Conf. Fuzzy Systems (San Diego, CA) Mar. 1992. – P. 1163–1170.

4. *Wang L.-X., Mendel J. M.* Fuzzy basis functions, universal ap- proximation, and orthogonal least-squares learning // IEEE Trans. on Neural Networks. – 1992. – Vol. 3. – P. 807–814.

5. *Натансон И. П.* Конструктивная теория функций. – М. : Государственное технико-теоретическое изд-во, 1949. – 688 с.

6. *Гончаров В. Л.* Теория интерполирования и приближения функций. – М. : Гос. техн.-теорет. изд-во, 1934. – 316 с.

7. *Lorentz G. G.* Bernstein polynomials. – Toronto: University of Toronto Press, 1953. – 130 p.

8. Потапов М. К. О приближении алгебраическими многочленами в интегральной метрике с весом Якоби // Вестник Московского университета. Сер. Математика и механика. – 1983. – Вып. 4. – С. 43–52.

9. *Дзядык В. К.* Введение в теорию равномерного приближения функций полиномами. – М. : Наука, 1977. – 512 с.

10. *Шведов А. С.* Нечеткое математическое программирование: краткий обзор // Проблемы управления. – 2017. – № 3. – С. 2–10.

Золотухин Алексей Евгеньевич – магистрант 2-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: ZolotukhinAlexey@yandex.ru

Леденева Татьяна Михайловна (научный руководитель) – д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: ledeneva-tm@yandex.ru

УДК 51-77

К ВОПРОСУ РОБОТИЗАЦИИ БИЗНЕС ПРОЦЕССОВ НА ПРЕДПРИЯТИИ

Е. С. Зяблова

Воронежский государственный университет

Введение

По мере структурирования компаний все более популярными становятся современные автоматизированные системы поддержки управленческой деятельности, так называемые, ERP-системы (от Enterprise resource planning – Управление ресурсами предприятия), которые позволяют создать единую среду для автоматизации планирования, учета, контроля и анализа всех основных бизнес-операция в масштабе предприятия. ERP система представляет собой комплексную информационную систему управления информацией внутри организации, решающую весь комплекс задач управленческого, регламентированного и других видов учета, в отличие от специализированного программного обеспечения, предназначенного для автоматизации конкретного бизнес-процесса или направления деятельности.

Конкуретноспособность и успешность компаний в условиях рыночной экономики напрямую зависит от их способности к быстрой адаптации и мгновенному реагированию на изменение рыночной среды. Новые горизонты бизнеса и новые задачи требуют частого пересмотра бизнес-процессов, а увеличивающиеся объемы накапливаемых данных – новых, более совершенных средств управления ими. Система управления предприятием позволяет топ-менеждменту и собственникам бизнеса повысить конкурентноспособность и инвестиционную привлекательность компании, за счет оптимизации и стандартизации бизнес-процессов с использованием лучших мировых практик и обеспечения прозрачности операционной и финансовой деятельности и применения IT-инструментов. Использование инструментов ERP – системы предоставляет возможность управлять компанией на основе всегда достоверной и актуальной информации для принятия взвешенных рещений [3].

Комплексная автоматизация управления предприятием – это не просто создание информационной системы, а переход к управленческой концепции, включающей в себя одновременно инструменты и стандарты управления.

Тема роботизации и автоматизации производства на сегодняшний день является одной из ключевых в развитии организации. Каждый день мы видим сообщения о разработке новых технологий и решений, которые предлагается реализовать на предприятиях для увеличения производительности и эффективности производства [1].

Человеческий ресурс является крайне ценным, чтобы позволить его расходовать на те направления, где машины справляются лучше, недопуская «человеческих» ошибок. Любой рутинный труд, выполняемый человеком, должен и может быть автоматизирован, чтобы люди могли сфокусироваться на других задачах, которые роботизированные системы пока не в состоянии выполнить.

Существует эффективное и современное решение для автоматизации процессов – роботизация рабочих процессов. Robot Process Automation (RPA) – это специализированные программные средства, которые имитируют работу пользователя, взаимодействуя с различными информационными системами через стандартный пользовательский интерфейс. Благодаря этому отпадает необходимость в трудоемкой и затратной программной интеграции и обеспечивается совместная работа нескольких корпоративных систем.

Роботов используют, если действия в системе четко определены, основаны на правилах и не требуют от сотрудников принятия нестандартных решений. Такие задачи можно поручить

роботам, чтобы увеличить скорость выполнения работы и уменьшить число сотрудников. Роботы избавят людей от монотонной работы, ошибок из-за снижения внимательности и помогут освободить время для творческих задач.

Общими признаками, объединяющими операции, которые имеют наибольшее шансы быть роботизированными эффективно, являются нижеследующие [2]:

- Регулярно и достаточно часто повторяются. Для того, чтобы любая автоматизация была эффективной, необходимо частотное повторение процесса. Это справедливо и для RPA;
- Важные для бизнеса компании. Действительно, никто не будет тратить деньги на малозначащие операции. При этом процесс может повторяться и не столь часто, однако быть весьма важным для компании, например, сбор данных для отчетности к еженедельному совещанию совета директоров;
- Требуют обработки значительных объемов данных. Это как раз тот аспект, который показывает преимущества роботизации. При возрастании нагрузок человек склонен совершать все больше ошибок, в то время как робот будет продолжать стабильную работу;
- Используют строгие бизнес-правила. Роботизированные процессы по определению предполагают исключение человека из принятия решения о выборе варианта исполнения процесса;
- Требуют работы не менее чем с одной электронной системой (внешней, внутренней). Здесь, конечно, имеется определенная тавтология. Робот обязан взаимодействовать с той или иной корпоративной системой для того, чтобы выполнить свою работу.

Постановка задачи

Перед нами стоит цель автоматизировать процесс разнесения платежей в SAP, тем самым освободить человеческие ресурсы в размере 4 человек.

Методом декомпозиции целесообразно выделить следующие частные подзадачи:

- Унификация процесса перед автоматизацией;
- Редактирование должностных инструкций;
- Подготовка BRD;
- Обучение сотрудников.

Главным эффектом от внедрения RPA ожидается повышение удовлетворенности сотрудников своей работой, поскольку им больше не придется ежедневно выполнять однообразные операции, и онис могут заниматься решением задач, соответствующих уровню их профессиональной подготовки. К тому же, мы ожидаем, что внедрение роботизации приведет не к сокращению сотрудников, а к уменьшению текучести кадров и стабилизации численности, и все это – благодаря повышению уровня удовлетворенности сотрудников своей работой.

Литература

1. *Кожухова, О. А.* Внедрение и использование ERP-систем на предприятии / О. А. Кожухова. – Красноярск : СГАУ, 2011. – 448 с.

2. Робот вместо человека: почему бизнесу важно внедрять RPA? TADетали. – Режим доступа: https://zen.yandex.ru/media/tadviser/robot-vmesto-cheloveka-pochemu-biznesu-vajno-vnedriat-rpa-tadetali-5c0f6a5944c73500ae93ae92 (дата обращения: 10.04.2021).

3. ERP система: статья. – Режим доступа: https://wiseadvice-it.ru/o-kompanii/blog/articles/erp-sistemy/ (дата обращения: 10.04.2021).

Зяблова Екатерина Сергеевна – студентка 2-го курса кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: igolkina.ft@yandex.ru

Иванкин Михаил Петрович (научный руководитель) – канд. тех. наук, доцент кафедры ERP-систем и бизнес процессов Воронежского государственного университета.
E-mail: mivankin@gmail.com

УДК 004.421

РАЗРАБОТКА ПРОЦЕДУРЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ЭЛЕМЕНТОВ ОБЪЕКТА СТРАНИЦЫ ДЛЯ ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ

М. О. Корнеев

Воронежский государственный университет

Введение

В настоящее время разработка веб-приложения проходит в рамках коротких по времени циклов, поэтому тестировщики подобных приложений, как правило, мало времени уделяют качеству автоматизированных end-to-end тестов. По мере быстрого развития приложений становится всё сложнее поддерживать тесты в актуальном состоянии. В результате чего часто происходит отказ от автоматизированных тестов, которые отлично справляются с обнаружением регрессий в приложениях.

Для облегчения поддержки тестовых скриптов применяется паттерн проектирования для автоматизированного тестирования Page Object [1]. Он заключается в создании для каждой страницы тестируемого приложения отдельного объекта, методы которого инкапсулируют логику работы с элементами пользовательского интерфейса. Применение паттерна также позволяет уменьшить количество кода путём отделения логики выполнения тестов от их реализации. Однако при частых изменениях пользовательского интерфейса необходимо также часто вручную создавать и обновлять элементы объектов страниц для поддержания работоспособности тестов.

Рассматриваемая проблема заключается в больших затратах на изменение элементов объектов страниц при поддержке тестовых скриптов. Для решения данной проблемы необходимо разработать ПО, которое способно автоматически генерировать элементы объекта страницы для тестирования веб-приложений.

Существует несколько готовых решений, среди которых SWD Page Recorder [2] и Selenium Page Object Generator [3]. Однако они имеют существенные недостатки:

– обработка только одной страницы за единицу времени;
– необходимость писать вручную понятные названия элементов;
– работа только со статическим контентом страницы;
– отсутствие генерации методов для работы с веб-элементами.

Для того, чтобы исправить данные недостатки в разрабатываемом ПО, были сформулированы следующие требования к проекту:

– возможность обработки нескольких страниц приложения одновременно;
– создание элементов с названиями, отражающими их суть;
– поддержка динамического контента;
– генерация методов для взаимодействия с веб-элементами.

1. Общее описание разрабатываемой процедуры

Разрабатываемый подход автоматической генерации элементов страницы состоит из шагов, представленных на рис. 1. Сначала создаётся модель тестируемого приложения путём обратного проектирования с помощью поискового робота, или так называемого веб-краулера [4]. Затем схожие страницы собираются в синтаксически и семантически значимые группы или кластеры. Модель на основе событий в виде графика вместе с дополнительной информа-

цией (например, DOM-документы и кластеры) подвергаются анализу с точки зрения объектов состояний, в результате чего получается объектно-ориентированная модель состояний. На последнем этапе данная модель преобразуется в объекты страницы с соответствующими названиями в виде текста на языке Java.

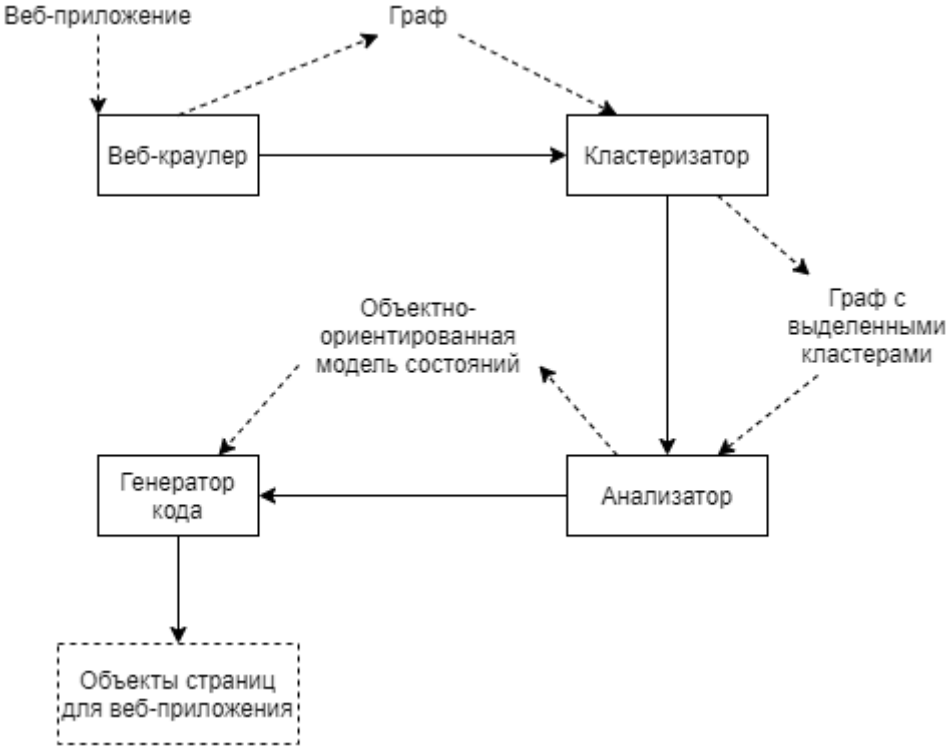


Рис. 1. Алгоритм выполнения процедуры

Таким образом, процедура состоит из работы 4 компонентов: веб-краулера, кластеризатора, анализатора и генератора кода. На вход будет подаваться веб-приложение вместе со всеми необходимыми данными (например, логины и пароли, текст для ввода). На выходе должен быть получен набор элементов объекта страницы, использующий фреймворк Selenium WebDriver [5] и организованный с помощью паттерна Page Factory [6].

Элементы объекта страницы генерируются автоматически, однако возможно проведение некоторой ручной работы для доработки полученного кода.

2. Детальное описание каждого этапа процедуры

2.1. Веб-краулер

На первом этапе работы программы необходимо получить высокоуровневое представление тестируемого приложения в виде объектно-ориентированной модели динамических состояний DOM, то есть в виде динамического графа состояний. Веб-краулер автоматически строит граф, включающий в себя динамические состояния DOM и переходы между ними. На рис. 2 показана часть возможного графа. После работы инструмент также возвращает следующую информацию о каждом динамическом состоянии: URL, список кликабельных элементов страницы, DOM-документ, скриншот страницы, список ссылок на другие состояния.

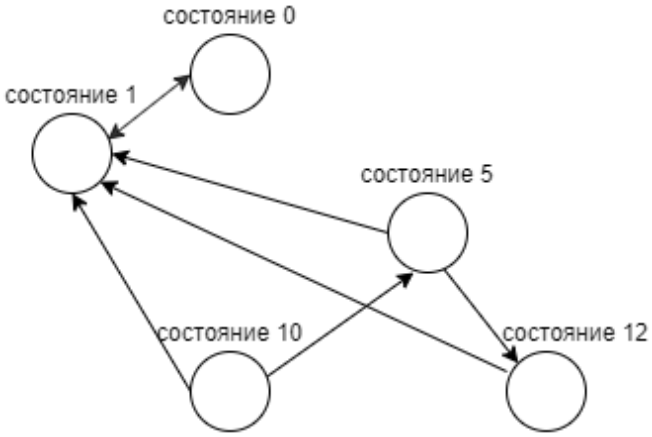


Рис. 2. Часть полученного графа

2.2. Кластеризатор

Анализ полученных от веб-краулера результатов вручную может быть чрезвычайно сложным для тестировщика. Количество динамических состояний и переходов между ними может быть велико, так как любое изменение в графическом интерфейсе страницы влечёт за собой создание нового динамического состояния.

Для решения этой проблемы будет использоваться кластеризатор. Он группирует в один кластер те страницы приложения, которые должны рассматриваться в рамках одного объекта страницы. Для этого был выбран иерархический агломеративный алгоритм, опирающийся на понятие сходства между веб-страницами.

Для сравнения страниц будут браться следующие характеристики: частота появления тэга, частота появления слова, URL и DOM-документ. Кластеризатор автоматически создаёт матрицу синтаксических признаков на основе каждой страницы. С помощью алгоритма, использующего полученные матрицы, рассчитывается сходство страниц, в результате чего возвращается набор кластеров, как показано на рис. 3.

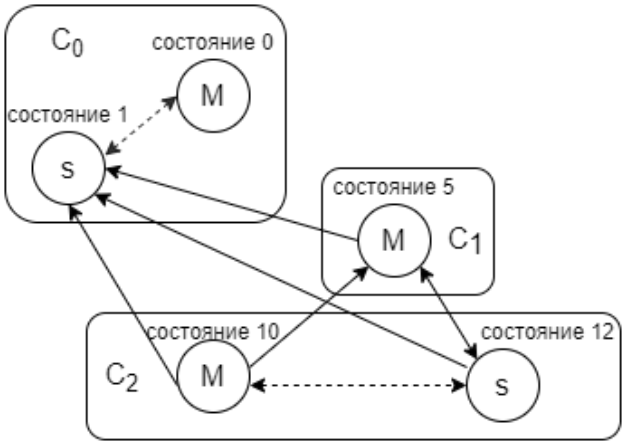


Рис. 3. Часть графа с выделенными кластерами

2.3. Анализатор

Работа данного модуля заключается в создании абстракции приложения с точки зрения объектов состояний. Процесс разделён на 3 части: расчёт разницы между DOM-документами, модификация графа, создание объектно-ориентированной модели состояний.

2.3.1. Расчёт разницы между DOM-документами внутри кластера

На данном этапе для каждого из полученных кластеров определяются главное и подчинённые состояния. Главное состояние является представлением всех остальных, соответственно, подчинённых состояний. Происходит выбор уникальной страницы внутри кластера в качестве наилучшего кандидата для объекта страницы, чтобы объединить все подчинённые состояния в рамках главного. В качестве главного берётся первое динамическое состояние, возвращённое веб-краулером. Далее происходит сравнение DOM каждого из подчинённых состояний с главным, при этом собирается информация о динамических частях страницы.

2.3.2. Модификация графа, основанного на кластерах

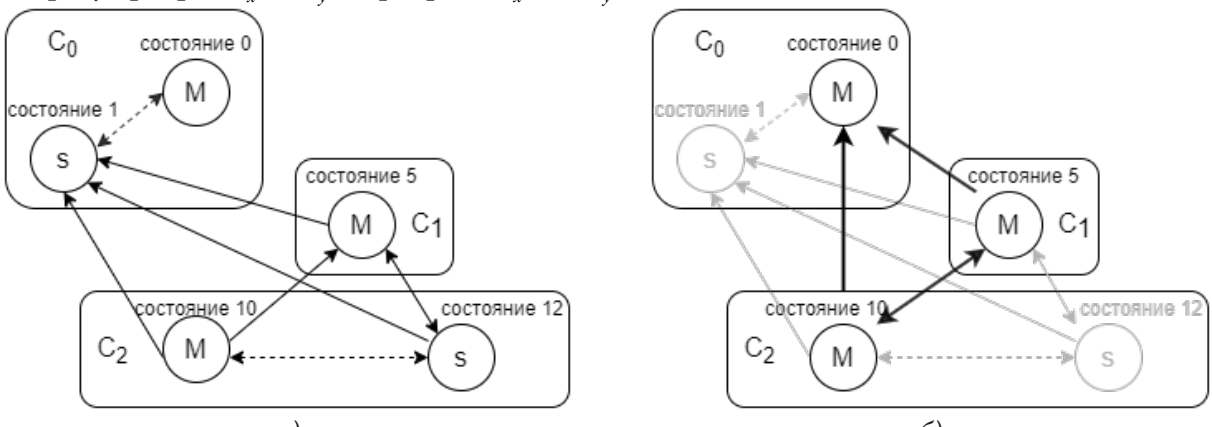
Данный этап отвечает за изменение графа, полученного от веб-краулера, путём добавления в него новой информации о кластерах, а также о главном и подчинённых состояниях. На изображении 4 представлен пример вида графа до (рис. 4а) и после внесения в него изменений (рис. 4б). На рис. 4а изображены три кластера из пяти страниц (состояний): $C_0 = \{состояние\ 0,\ состояние\ 1\}$, $C_1 = \{состояние\ 5\}$ и $C_2 = \{состояние\ 10,\ состояние\ 12\}$. Главное состояние помечено буквой «*M*» (master), подчинённые – буквой «*s*» (slave). Пунктирные стрелки представляют рёбра внутри кластера, сплошные – соединения между кластерами. На рис. 4б представлен граф в преобразованном виде: рёбра внутри кластера удаляются, так как включённые страницы образуют уникальный объект страницы, рёбра между кластерами изменяются в соответствие с алгоритмом, описанным ниже. Положим, есть кластеры C_x и C_y:

1) ребро $(M_x \to M_y)$: ребро между двумя главными состояниями (напр., состояние 10 и состояние 5), изменений не происходит;

2) ребро $(M_x \to s_y)$: ребро между главным состоянием одного кластера и подчинённым состоянием другого. Случай кластеров C_0, C_2 и ребра между состоянием 10 и состоянием 1. Так как состояние 1 – подчинённое, то направление ребра должно быть изменено так, чтобы оно входило в главное состояние (состояние 0). Поэтому, ребро $(M_x \to s_y)$ становится ребром $(M_x \to M_y)$;

3) ребро $(s_x \to M_y)$: ребро между подчинённым состоянием одного кластера и главным состояние другого. Случай кластеров C_1, C_2 и ребра между состоянием 5 и состоянием 12. Так как состояние 12 – подчинённое, направление ребра должно быть изменено так, чтобы оно выходило из главного состояния кластера (состояние 10). Поэтому ребро $(s_x \to M_y)$ становится ребром $(M_x \to M_y)$;

4) ребро $(s_x \to s_y)$: ребро между подчинёнными состояниями двух кластеров (напр., состояние 12 и состояние 1). В этом случае главные состояниях данных кластеров соединяются, преобразуя ребро $(s_x \to s_y)$ в ребро $(M_x \to M_y)$.



а) *б)*

Рис. 4. Модификация графа а) исходный граф, б) изменённый граф

2.3.3. Создание объектно-ориентированной модели состояний

На данном этапе имеется абстрактное представление приложения в виде графа, кластеры и различия между DOM. Эта информация обрабатывается анализатором для создания объектно-ориентированной модели состояний приложения. Для каждого из динамических состояний создаётся абстракция объекта страницы следующим образом:

1) URL обрабатывается и обрезается, в результате чего получается осмысленное название класса для абстракции объекта страницы;

2) веб-элементы, которые веб-краулер инициировал как события, добавляются в качестве экземпляров WebElement в абстракцию объекта страницы;

3) переходы между состояниями сопоставляется с функциональностью типа «Navigation»;

4) DOM-документы анализируются на наличие форм, которые сопоставляются с функциональностью типа «Action»;

5) различия между DOM-документами сопоставляются с функциональностью «Getter». Это выполняется только для главных состояний.

Затем происходит объединение нескольких полученных абстракций в рамках кластера в одну абстракцию объекта страницы.

2.4. Генератор кода

На последнем этапе происходит преобразование объектно-ориентированной модели состояний в код на языке Java для фреймворка Selenium WebDriver. Для каждой абстракции (то есть, кластера) выполняются следующие действия:

1) создание класса Java с именем, полученным от анализатора, и необходимыми импортами для Selenium WebDriver;

2) создание экземпляров WebElement для каждого веб-элемента с аннотацией @FindBy и соответствующим локатором;

3) создание конструктора по умолчанию для управления браузером. Конструктор использует паттерн PageFactory, чтобы инициализировать все веб-элементы;

4) создание метода типа «Navigation» для каждого перехода из главного состояния текущего кластера в состояния других кластеров;

5) создание одного или нескольких методов типа «Action»;

6) создание метода типа «Getter» для каждого различия внутри кластера.

Заключение

В статье описана разработанная процедура автоматической генерации элементов объекта страницы для тестирования веб-приложений. Приложение, реализующее данную процедуру, должно состоять из четырёх модулей: веб-краулер, кластеризатор, анализатор и генератор кода. Данный подход поможет сократить время на создание и изменение элементов объектов страниц при написании и поддержке тестовых скриптов.

Литература

1. Selenium и Page Object паттерн: http://internetka.in.ua/selenium-page-object/ (дата обращения: 14.04.2021).

2. SWD Page Recorder: записывает PageObject-классы для Selenium WebDriver: https://habr.com/ru/post/191802/ (дата обращения: 15.04.2021).

3. Selenium Page Object Generator: https://github.com/rickypc/selenium-page-object-generator/ (дата обращения: 15.04.2021).
4. О веб-краулерах: http://crawlers.info/pages/crawlers.html/ (дата обращения: 17.04.2021).
5. Selenium WebDriver: https://www.selenium.dev/ (дата обращения: 19.04.2021).
6. Selenium: работаем с элементами страницы, используя @FindBy и PageFactory: https://habr.com/ru/post/134462/ (дата обращения: 20.04.2021).

Корнеев Максим Олегович – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: korneev.maxim97@yandex.ru

Борисенков Дмитрий Васильевич (научный руководитель) – канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: xuser@relex.ru

УДК 004.42

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ НА ПЛАТФОРМЕ ANDROID ДЛЯ ЗДОРОВОГО ОБРАЗА ЖИЗНИ

А. С. Костина

Воронежский государственный университет

Введение

В современном мире очень стремительно растет потребность в использовании мобильных устройств. Всего пару лет назад заказывали товары или искали необходимую информацию на сайтах, а сейчас делаем это все в мобильных приложениях. С развитием технологий появляется потребность и в других жизненных сферах. Например, здоровое питание и спорт.

На данный момент здоровый образ жизни стал важнейшей частью жизни современного человека. С каждым днем всё больше и больше людей начинают заниматься йогой, футболом, фитнесом и другими активностями. А также уделяют особое внимание своему дневному рациону. Фитнес сообщество развивается очень быстро, так как мы часто видим рекламу в интернете или соответствующие статьи в журналах. На данный момент, даже фитнес клубы создают свои собственные приложения, где клиент может записаться на тренировку, почитать новости о клубе или другую необходимую информацию. Однако, большинство из нас нуждается в таком приложении, где можно отслеживать не только свою физическую активность, но и дневной рацион.

1. Анализ функциональности и разработка приложения

Перед началом разработки был проведен анализ существующих приложений и обратной связи от их пользователей.

На данный момент, большинство приложений реализовано для платформы IOS, поэтому существует необходимость в приложении для пользователей других платформ. Для реализации была выбрана платформа Android на Flutter с использованием языка Dart, так как эта среда разработки позволяет создать продукт для двух платформ сразу.

В результате анализа были выделены две роли использования приложения: пользователь и администратор. Для наглядного представления были созданы две Use Case диаграммы.

Пользователь имеет возможность составлять свою тренировку из базовых упражнений. Добавлять свои рецепты или использовать имеющиеся (рис. 1).

Рис. 1. Диаграмма вариантов использования для пользователя

Администратор управляет всем приложением, имея возможность добавлять новые роли и функциональности (рис.2).

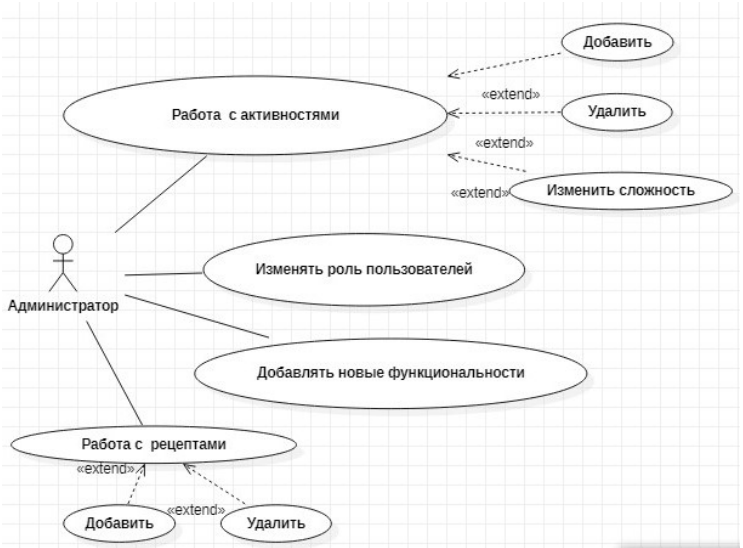


Рис. 2. Диаграмма вариантов использования для администратора

В процессе разработки были созданы соответствующие страницы:

1. Main.dart – данная страница является входной точкой в приложение. Здесь определяется, какая страница будет запускаться первой. В нашем случае это страницы регистрации.

2. Auth.dart – страница, в которой реализована возможность регистрации и синхронизация данных с Firebase.

3. Workouts-list.dart – на данной странице представлен функционал работы с тренировками. Здесь описаны соответствующие поля для тренировок и их фильтрация.

4. User.dart – страница для работы с пользователями.

5. Auth-screen.dart – страница, где описаны регистрация и вход в приложение. А также соответствующие проверки (например корректность логина, нахождение пользователя в базе и соответствие пароля).

Для наглядного представления была создана ClassDiagram(Рис.3)

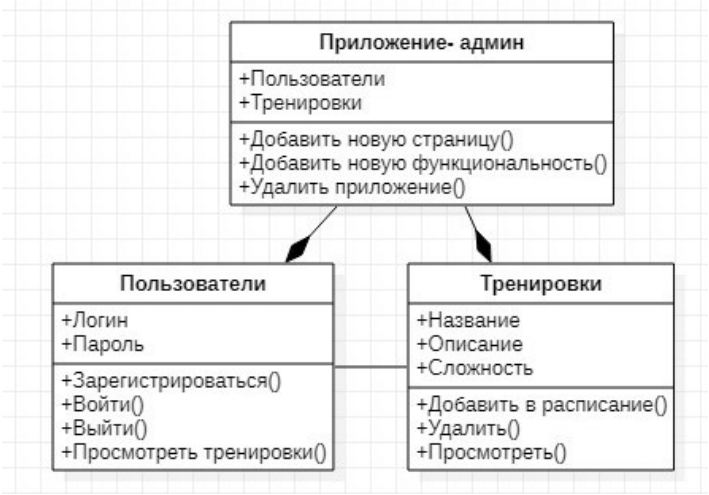
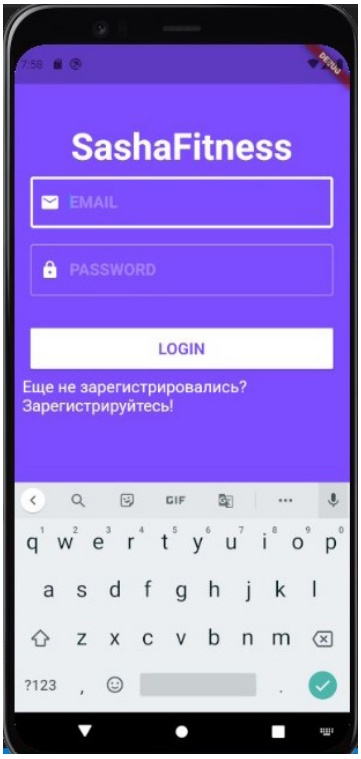


Рис. 3. Диаграмма классов

Скриншоты по страницам представлены на соответствующих рисунках (рис. 4-6).



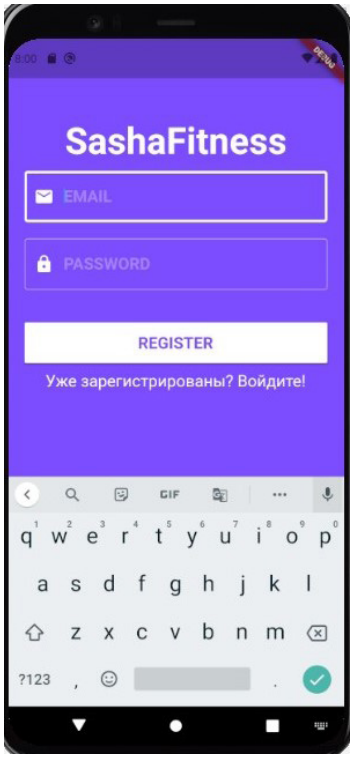


Рис. 4–5. Страницы входа и регистрации в приложении

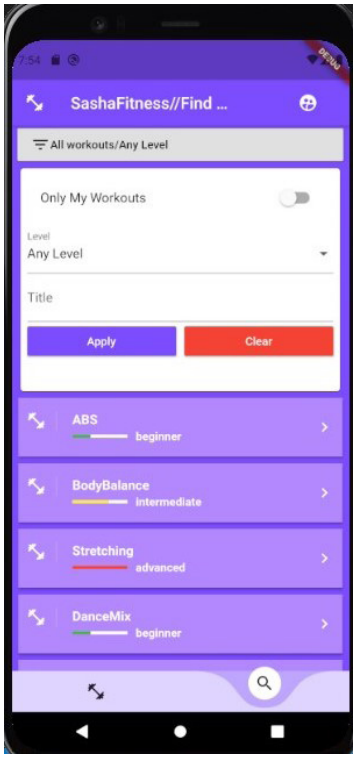


Рис. 6. Страница для просмотра существующих тренировок по фильтру

Рабочая область приложения состоит из отдельных элементов(виджетов), каждый из которых имеет свое значение. Например, возможность фильтрации тренировок, выбор плана питания или же переход на другой экран.

Для навигации, работы виджетов и некоторых функциональностей используются специальные библиотеки, которые подключаются на странице pubspec.yaml.

1. Например:
2. Material. Dart – стандартная библиотека для использования виджетов.
3. Provider – библиотека для управления состоянием.
4. Fluttertoast – библиотека для работы со всплывающими окнами.
5. Firebase_auth – библиотека для работы с базой данных.
6. Cloud_firestore – библиотека для хранения данных.

Для хранения данных о пользователе, тренировок и питания используется база данных Firebase, которая сразу же синхронизирует изменения данных.

Использование одного приложения для разных действий помогает пользователю экономить время и хранить информацию в одном месте.

Современные технологии помогают ускорить процесс разработки, поэтому в данной работе была использована среда Flutter, которая даёт возможность разработать одно приложение для двух платформ: IOS и Android.

При создании продукта возникает сложность, чтобы приложение не было перегружено, тем самым рассеивая внимание пользователя.

Поэтому, чтобы реализовать данное приложение, необходимо собрать информацию от пользователей, какое приложение хотели бы использовать они, собрать статистику о видах физической активности, которые актуальны для пользователей. А также сделать это приложение таким, чтобы оно побуждало и других больше времени уделять здоровому образу жизни.

Заключение

В настоящее время реализована работа с тренировками: возможность самостоятельной регистрации в приложении, работа с тренировками (добавление, удаление, изменение), получение данных из базы данных, отображение тренировок по заданному признаку (сложность, название, авторство), возможность составления авторских программ для тренировок.

В ближайшее время планируется для реализации работа с рецептами, а именно: возможность самостоятельного добавления рецептов, а также работа с существующими, просмотр стандартных рецептов или от других пользователей и добавление их в свою библиотеку.

Данный продукт может быть актуальным для тех пользователей, которые отводят здоровому образу жизни большое количество времени.

Литература

1. Уроки по Flutter.- URL: https://flutter.su/ (дата обращения: 29.01.2021)

2. Учусь Firebase.- URL:https://riptutorial.com/Download/firebase-ru.pdf (дата обращения: 5.02.2021)

3. Firebase. – URL: https://firebase.google.com/ (дата обращения: 10.02.2021)

4. Крис Бакетт. Dart в действии – 2016, 530с

5. Flutter. Documentation – URL: https://flutter.dev/docs (дата обращения 2.03.2021)

Костина Александра Сергеевна – студентка 4-го курса кафедры МО ЭВМ Воронежского государственного университета. E-mail: sasha.kostina.99@mail.com

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры МО ЭВМ Воронежского государственного университета.
E-mail: Bolotova.svetlana@gmail.com

УДК 681.3.06

ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ИНТЕРНЕТ-ТЕХНОЛОГИЙ ВО ВНЕУЧЕБНОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ ВОРОНЕЖСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

И. Д. Коток

Воронежский государственный университет

Введение

С приходом коронавирусной инфекции в 2020 году, очень сильно изменилось отношение студентов к различного рода мероприятиям, многие из которых практически полностью перешли на онлайн формат.

Разработка приложения для проведения заочного этапа кубка ПММ по интеллектуальным играм стала, как никогда, актуальной задачей [4–6]. Ранее разработанное приложение было предусмотрено в основном для хранения вопросов, формирования их базы и использования её студентами для тренировок, также там был предусмотрен режим турнира, где студенты могли пройти отбор для участия в очном этапе мероприятия.

В Воронежском государственном университете достаточно хорошая подготовка у студентов, которые достойно представляют вуз на чемпионатах по интеллектуальным играм различного уровня. На базе Объединенного Совета обучающихся сформирован клуб интеллектуальных игр ВГУ. Для развития этого студенческого сообщества было решено доработать приложение и добавить в него несколько новых режимов, которые смогут помочь студентам в подготовке и организации турниров на базе Воронежского государственного университета.

1. Разработка интерфейса приложения

После того, как были созданы основные контроллеры необходимо было подумать об интерфейсе приложения. Для того, чтобы пользователь смог взаимодействовать с ним было необходимо создать представления – файлы с разрешением cshtml. В данном случае, представления не являются html-страницей, так как мы имеем наличие кода, который был написан на языке C# [1–2].

Рассмотрим представление новостной ленты. Основные стили для всех страниц прописаны в _Layout.cshtml, данный файл автоматически подключается при создании нового представления. Остается только добавить содержимое для текущей страницы.

Для того, чтобы можно было использовать пагинацию страниц и расположение на ней необходимых объектов необходимо подключить модель, которую будет использовать представление, в этом случае это PagedList.IPagedList<IntGame.Models.News.News>.

После того, как контроллер передаст модель, используя цикл foreach, происходит проход по всему списку, который содержит модель. Получаемая информация выводится на страницу. Для вывода информации используется строго типизированный шаблонный хелпер DisplayFor, который достает из модели запрашиваемую информацию, используя лямбда выражения.

Пример добавления кнопок для удаления или изменения вопроса можно пронаблюдать в листинге 1.

Исходя из логики приложения, удалять и изменять новости может только пользователь, который имеет роль «admin». Остальные пользователи не смогут увидеть данные кнопки, так как для них они будут скрыты.

Листинг 1

Метод PlayGameSecondStep

```
@if (User.IsInRole(«admin»))
        {
            <div class=»container»>
                <div class=»col-md-1 col-md-offset-4»>
                        <p><a class=»btn btn-default» href=»/News/DeleteNews/@item.
NewsId»>Удалить</a></p>
                </div>
                <div class=»col-md-1 col-md-offset-1»>
                 <p><a class=»btn btn-default» href=»/News/NewsEdit/@item.NewsId»>Из-
менить</a></p>
                </div>
            </div>}
```

Пример работы этого представления продемонстрирован на рис. 1.

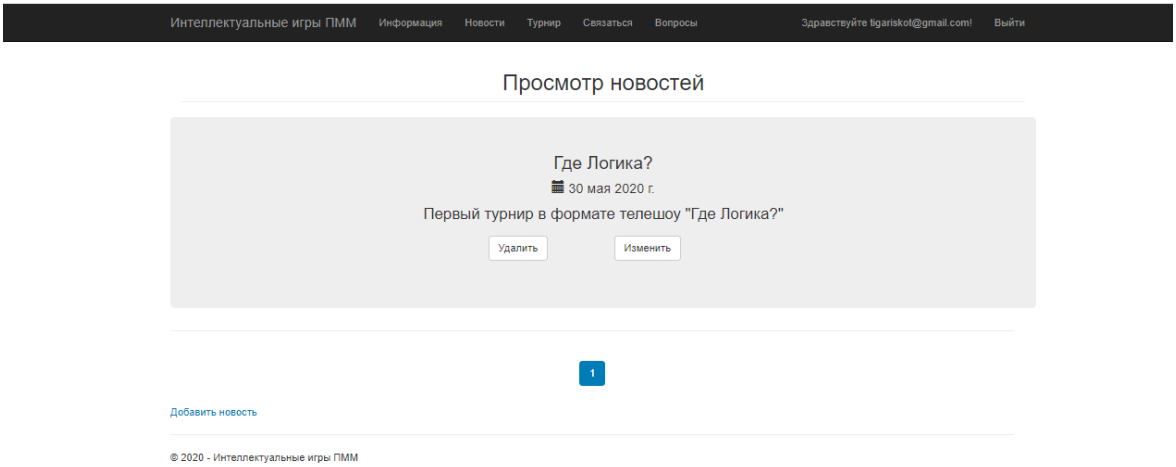


Рис. 1. Работа представления NewsList

Работа всех представлений аналогична рассматриваемому ранее. Отличаться они будут только использованием различных хелперов, которых требует конкретная ситуация. Для работы со стилями представлений была выбрана библиотека Bootstrap, которая реализует все необходимые возложенные на нее цели.

Заключение

Таким образом, клуб интеллектуальных игр ВГУ может использовать предыдущую версию приложения, для подготовки своих членов к соревнованиям и проведения собственных мероприятий. К сожалению, текущий формат будет удобен для узкого количества игр, сейчас основной уклон направлен на популярную телевизионную игру «Что? Где? Когда?».

В связи с этим, на данный момент рассматриваются два новых режима, такие как «Своя игра» и «Брейн-ринг». Каждый из режимов должен будет предусматривать видеосвязь участников и ведущих во время игры, чтобы максимально приблизить условия игры к реальным, исключить нарушение правил и сделать её более прозрачной.

Модификация имеющегося приложения поспособствует развитию студенческого самоуправления, которое направлено в сферу интеллектуальных игр. Также станет хорошей платформой для тренировок сборных команд, представляющих университет на соревнованиях данного направления. Увеличение количества режимов даст возможность студентам найти себя в новой сфере, поспособствует развитию креативного и критического мышления, улуч-

шению коммуникативных способностей студентов, поможет им раскрепоститься и адаптироваться в новой среде, особенно, если речь идет о первокурсниках [3].

Большая роль в организации, подготовке и проведении подобных мероприятий возлагается на отдел по воспитательной работе. Приложение облегчит эту работу и поможет ее систематизировать, добавит возможность узнать, какие темы наиболее интересны молодежи сейчас, выявит проблемные моменты и покажет новые подходы (точки соприкосновения) во взаимодействии студентов и администрации университета.

Литература

1. *Албахари Джозеф, Албахари Бен*. C# 7.0 Справочник. Полное описание языка. : Пер. с анг. – СПб. : ООО «Диалектика», 2019. – 1024 с.

2. .NET Documentation | Microsoft Docs. – Режим доступа: https://docs.microsoft.com/en-us/dotnet/

3. *Коток И. Д.* Разработка программного обеспечения для проведения интеллектуальных компьютерных игр студенческим советом факультета прикладной математики, информатики и механики // Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч.-техн. конф. (Воронеж, 11–13 ноября 2019 г.). – Воронеж: Издательство «Научно-исследовательские публикации», 2020. – С. 443–444.

4. *Коток И. Д.* Разработка программного обеспечения для проведения интеллектуальных викторин // Информационные технологии в образовательном процессе вуза и школы: материалы XIV Всероссийской научно-практической конференции (Воронеж, 25 марта 2020 г.). – Воронеж: Воронежский государственный педагогический университет, 2020. – С. 171–175.

5. *Коток И. Д.* Интернет-технологии в активизации работы студенческого совета факультета прикладной математики, информатики и механики // Математика, информационные технологии, приложения: сборник трудов Межвузовской научной конференции молодых ученых и студентов, Воронеж, 27 апреля 2020 г. – Воронеж: Научная книга, 2020. – С. 123–126.

6. *Коток И. Д.* Разработка программного обеспечения для работы студенческого объединения в сложной эпидемиологической обстановке / Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч.-техн. конф. (Воронеж, 7–9 декабря 2020 г.). – Воронеж: Издательство «Научно-исследовательские публикации», 2021. – С. 545–551.

Коток Игорь Дмитриевич – магистрант 1 курса кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики Воронежского государственного университета. E-mail: tigariskot@gmail.com

Ускова Ольга Фёдоровна (научный руководитель) – канд. техн. наук, профессор кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики Воронежского государственного университета. E-mail: nat-uskova@mail.ru

Гришаев Олег Викторович (научный руководитель) – канд. ист. наук, доц., заведующий кафедрой новейшей отечественной истории, историографии и документоведения исторического факультета Воронежского государственного университета, проректор по воспитательной и социальной работе Воронежского государственного университета. E-mail: grishaev@vsu.ru

УДК 004.92

АЛГОРИТМЫ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ ЛАНДШАФТОВ

Л. И. Лесных

Воронежский государственный университет

Введение

В настоящей статье рассматриваются алгоритмы процедурной генерации, то есть автоматического создания объектов, при котором для одних и тех же начальных параметров генерируется один и тот же объект. Для генерации ландшафта происходит генерация карты высот, то есть некоторого двумерного изображения. В данной работе будут рассмотрены некоторые из методов, позволяющих это сделать.

1. Генерация карты высот клеточными автоматами

Клеточные автоматы – дискретная модель, состоящая из равномерной сетки ячеек, каждая из которых может находиться в одном из состояний, заданных некоторым конечным множеством.

Примером клеточного автомата является игра «Жизнь». Основные правила этой игры: у клетки может быть два состояния, они (кроме начального) определяются количеством соседних живых клеток предыдущего шага.

Еще один клеточный автомат – Steppers. В нем у клетки может быть три состояния: неживая, молодая, старая. Новые состояния неживой и старой клеток определяются соседями, а молодая переходит в состояние старой.

Для получения карты высот можно задать какое-то случайное распределение начального поколения клеток и запустить игру. По прошествии некоторого количества итераций подсчитать для каждой ячейки, сколько раз изменялось состояние с неживой на живую и наоборот. По матрице, заполненной этими значениями, можно получить карту высот.

На рис. 1 изображены карты высот, полученные такими подсчетами. Здесь количество поколений равно 32, слева изображена карта игры «Жизнь», справа – Steppers. На левом изображении присутствует много пустых мест. Это происходит из-за того, что в этом клеточном автомате клетки могут достаточно быстро вымирать с течением времени. Для автомата Steppers вымирание происходит не так активно.

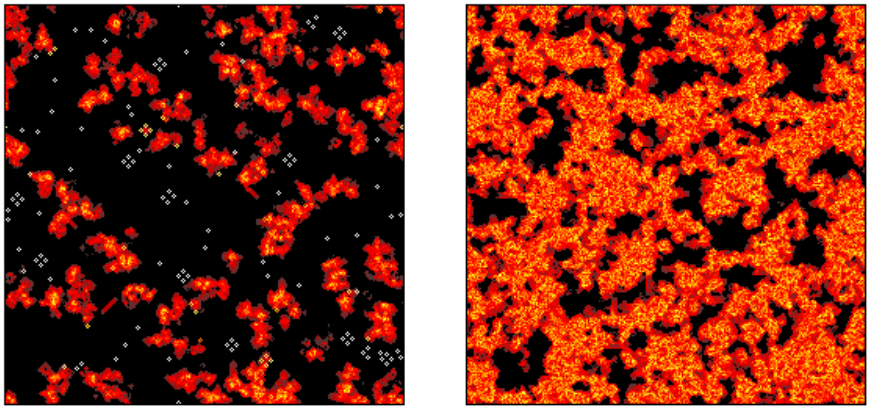


Рис. 1. Карты высот, полученные клеточными автоматами

2. Генерация карты высот при помощи симплекс-шума

Этот метод относится к так называемым градиентным шумам. Их идея состоит в том, чтобы создать сетку случайных градиентов, а затем интерполировать для получения значений в точках, лежащих между узлами.

Будем рассматривать только двумерный случай. В нем для симплекс-шума используется треугольная сетка. В узлах генерируются градиенты. Внутри ячейки значения получаются суммированием вкладов от значения в каждом узле, составляющем ячейку, умноженных на некоторую функцию затухания. Эта функция подбирается таким образом, чтобы она перед переходом к следующей ячейке достигала нуля. Это дает возможность вычислять значения внутри ячейки только по составляющим ее узлам.

Определить, в каком треугольнике находится произвольная точка пространства, можно в два этапа. Во-первых, входное координатное пространство наклоняется по главной диагонали (рис. 2).

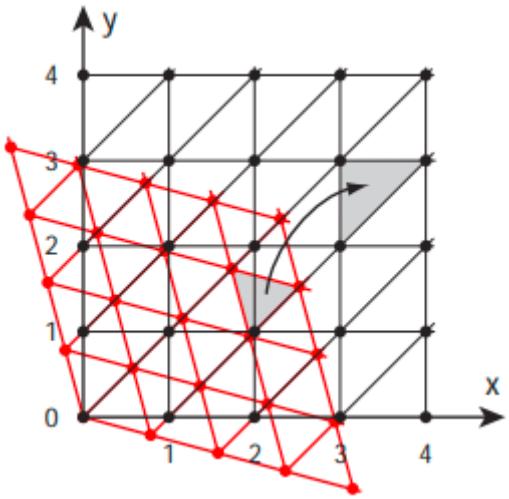


Рис. 2. Преобразование координатного пространства

Координаты точек при этом преобразуются по формулам:

$$x' = x + (x + y) \cdot F,$$
$$y' = y + (x + y) \cdot F,$$
$$F = \frac{\sqrt{3} - 1}{2}.$$

По целым частям преобразованных координат можно определить, в каком квадрате находится точка. Если абсцисса больше ординаты, то в нижнем треугольнике, иначе – в верхнем.

После этого на преобразованной сетке вычисляются значения внутри ячейки. Затем выполняется обратное преобразование координат:

$$x = x' - (x' + y') \cdot G,$$
$$y = y' - (x' + y') \cdot G,$$
$$G = \frac{3 - \sqrt{3}}{6}.$$

На рис. 3 изображена карта высот, сгенерированная при помощи симплекс-шума.

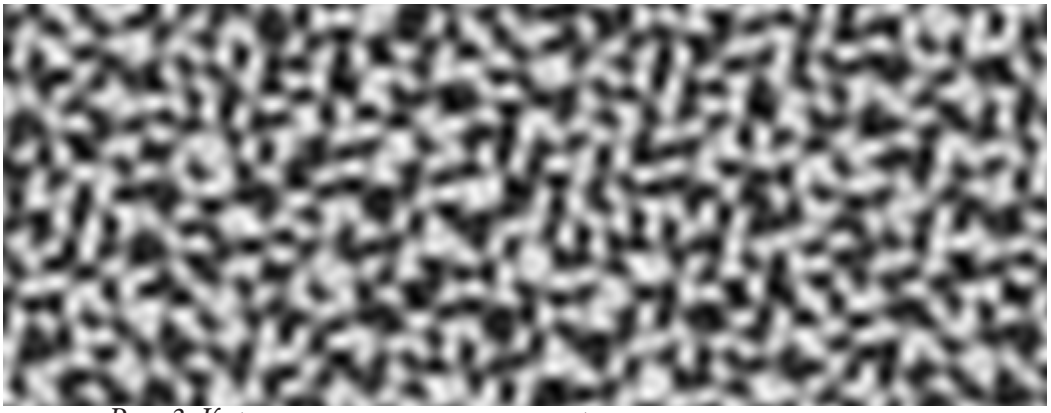


Рис. 3. Карта высот, полученная при помощи симплекс-шума

3. Применение карт высот к генерации ландшафтов

Если применять карты высот к генерации ландшафтов в чистом виде, получатся довольно скучные пейзажи, поскольку будут только перепады высоты. Для придания рельефу более реалистичной формы можно внутри каждой ячейки запустить генерацию еще раз и наложить ее на уже имеющуюся карту, но с меньшей амплитудой, это позволит «прорисовать» более мелкие детали, например, камни.

Заключение

В настоящей работе были рассмотрены некоторые методы процедурной генерации ландшафтов. Клеточные автоматы в некоторых случаях имеют свойство затухать, поэтому могут получиться почти пустые карты. Это видно по левому изображению на рисунке 1. Кроме того, для получения таких карт необходимо провести несколько итераций, что может негативно сказываться на производительности. Карта же, полученная симплекс-шумом выглядит более разнообразно и требует меньших вычислительных затрат.

Литература

1. *Dong, J.* Survey of Procedural Methods for Two-Dimensional Texture Generation / J. Dong, J. Liu, K. Yao, M. Chantler, L. Qi, H. Yu, M. Jian // Sensors, 2020. – Vol. 20. – 26 p.

2. Simplex noise. – Режим доступаhttps://en.wikipedia.org/wiki/Simplex_noise.

3. Simplex noise demystified. – Режим доступа: https://weber.itn.liu.se/~stegu/simplexnoise/simplexnoise.pdf.

4. Клеточный автомат Steppers. – Режим доступа: https://habr.com/ru/post/237629.

Лесных Любовь Игоревна – магистрант 2-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: torigatro@ktbk.ru

Корольков Олег Геннадьевич (научный руководитель) – канд. физ.-мат. наук, доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: korolkov@amm.vsu.ru

УДК ***

ПРОБЛЕМА СИНХРОНИЗАЦИИ ДАННЫХ МЕЖДУ ОТДЕЛЬНЫМИ НОДАМИ

П. С. Лесунов, Т. В. Анищев

Воронежский государственный университет

В связи с растущим числом сервисов параллельного обслуживания, с которыми одновременно работает большое число пользователей, возникает проблема, известная в сфере классических баз данных как «Dirty read».

Условно данную проблему можно продемонстрировать на следующем примере:

1) Пользователь_1 обновляет данные в файле Файл_1;

2) Пользователь_2 читает данные из файла Файл_1;

3) Система записывает изменения Пользователя_1 в Файл_1.

Пользователь_2 получает устаревшую информацию из-за задержек при обработке операций в СУБД.

Для обеспечения актуальности информации современные СУБД применяют ряд технологий [1]. Это может быть установка блокировок на определенные области данных (Locks), принудительная запись всех открытых транзакций в базу при попытке чтения из нее и т.д. (рис. 1).

На рис. 1 схематично представлена возникающая в условиях многопользовательского режима «ошибка двойного расходования».

Рис. 1. Ошибка двойного расходования

Ошибку «двойного расходования» наглядно демонстрирует следующий пример. Человек заходит в кафе и делает заказ стоимостью 500 рублей, наличные попадают в кассу кафе. Человек сможет потратить еще 500 рублей, но не те же самые 500 рублей, которые попали в кассу. При использовании цифровых денег возникает возможность копировать и перенаправлять, и таким образом, тратить дважды одни и те же средства.

В сфере здравоохранения, можно рассмотреть ситуацию двух клиник, в одной из которых пациент лечился по полису ДМС и в другую обратился по этому же полису, возникнет та же ситуация двойного расходования. Вторая клиника не сможет определить корректный статус полиса.

Рассмотренные проблемы возникают из-за того, что пользователи во время работы с документами могут быть подключен к разным серверам (рис. 2).

Решением данных проблем может являться привязка пользователя к конкретному серверу во время работы с документом. После подтверждения всех изменений, актуальная копия будет разослана по всем серверам в сети.

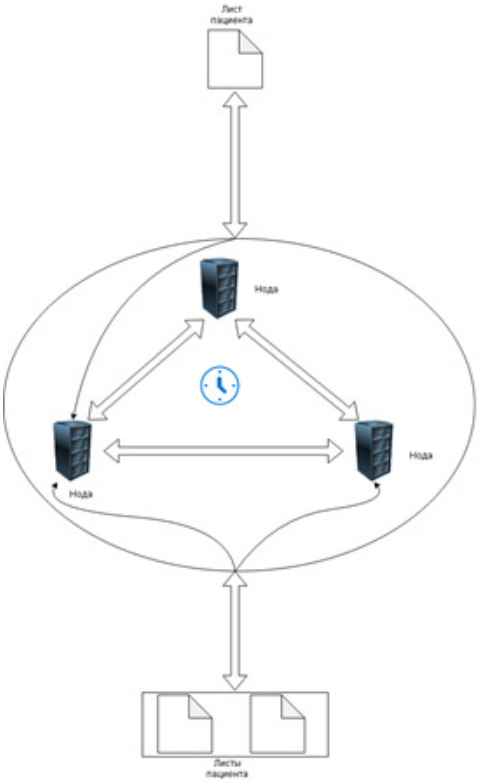


Рис. 2. Классический блокчейн

Соответственно любой другой пользователь, желающий работать с этим документом может быть подключен к любому другому серверу и работать с информацией последней редакции.

Дальнейшее изложение статьи будет спроецировано на описанную выше проблему ДМС в здравоохранении. Предполагается, что пользователи являются авторизованными клиниками, в момент выписки пациента его данные становятся доступны для всех клиник (рис. 3).

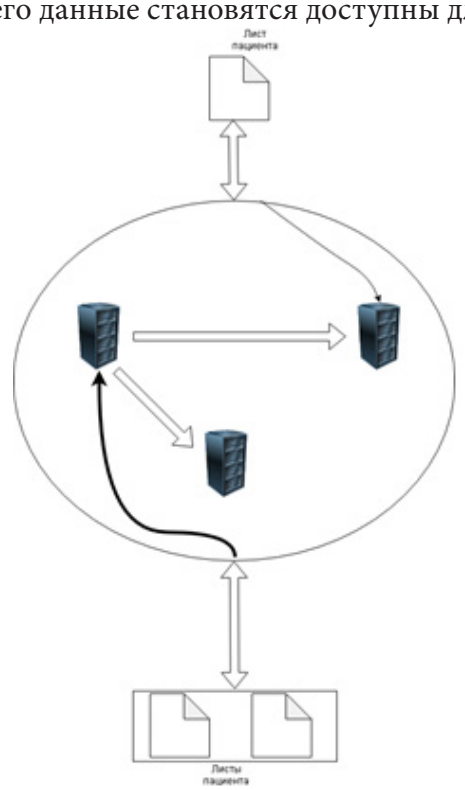


Рис. 3. Блокчейн с жесткой привязкой пользователя

Принимая во внимание теоретические возможности и предпосылки создания системы хранения данных в блокчейн – based DDB [2–3], можно отметить, что существует потребность создания системы, способной определять, когда пользователь подключился или отключился от конкретной ноды. Поскольку сильной стороной блокчейна является обеспечение неизменности внесенных данных, в рамках работы наиболее подходящим форматом выбрано медицинское заключение, действующее в правовом поле Российской Федерации.

В качестве распределенной среды распространения выбрана блокчейн – сеть Etherium (доступность и наличие системы обработки смарт-контрактов, как средства дополнительного расширения функционала (рис. 4)).

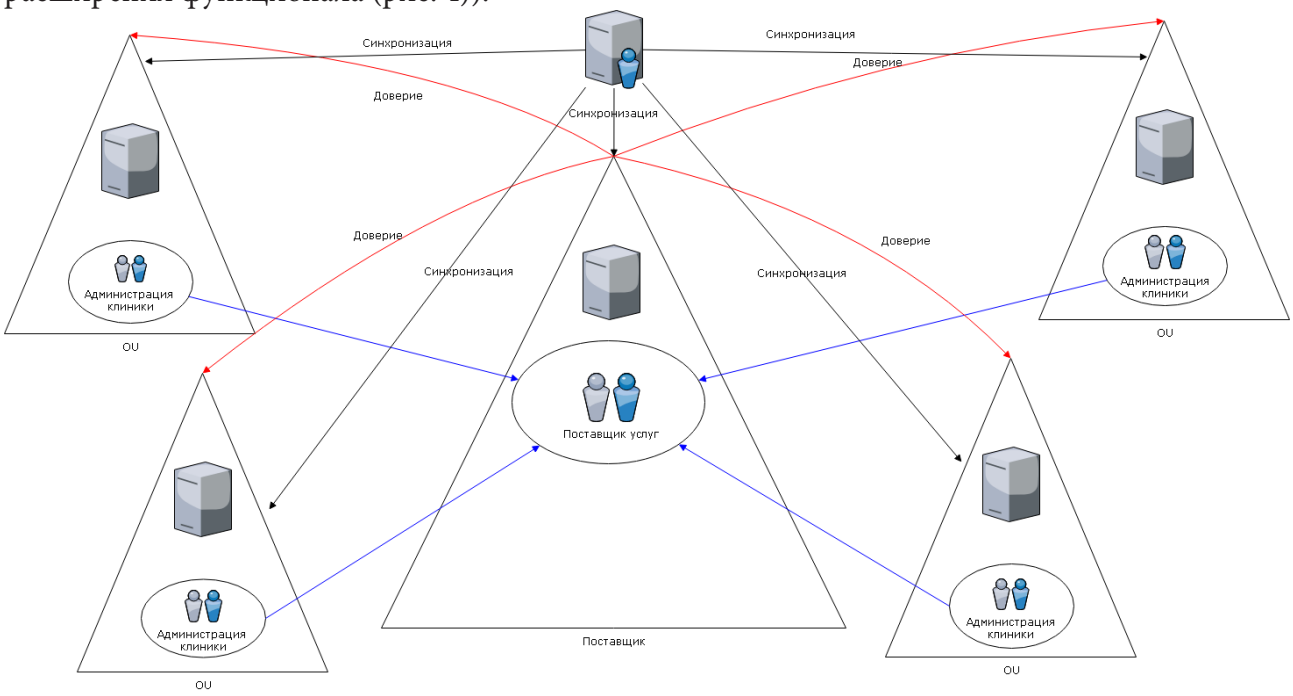


Рис. 4. Архитектура проекта

В представленной схеме отражено частное взаимодействие клиники и поставщика услуг. Для осуществления операции обмена/покупки (при реализации страхового полиса), в качестве посредника стороны могут выбрать любую ноду, вызывающую их доверие. Она может принадлежать банку или клиенту, или третьей стороне.

Главным достоинством такой системы является устранение проблемы «двойного расходования». Решение уже заложено при проектировании архитектуры [4].

В момент, когда пользователь подключается к любой ноде, сервер выполняет echo запрос, проверяя, подключен ли пользователь с таким ID уже к какой-либо ноде. Если нода, работающая в данный момент с этим ID получит этот запрос, она на него ответит. Соответственно, нода может однозначно определить статус любого ее пользователя.

Далее, нода подключается к соседней доверенной ноде и получает информацию о пользователе.

В течении сессии подключения вся информация о движениях, связанных с этим пользователем, сохраняется только на этой ноде. В момент отключения пользователя информация передается остальным нодам.

В текущей концепции хранения, история пользователя определяется набором транзакций. После окончательного подтверждения документа в соседних нодах он переходит в статус read-only. Жизненный цикл пользователя выглядит так:

– Подключение к выбранной ноде;

– Получение текущего баланса;

– ….
– Проведение N-транзакции;
– Отключение от ноды;
– Синхронизация с соседними узлами.

Поскольку для сферы здравоохранения неизменность уже внесенных данных критична, необходимо вести некое логирование всех записей с возможностью отката до любого описанного состояния.

Проще всего это сделать в текстовом виде. В качестве блоков, обрамляющих некие конечные изменения удобно выбрать тэги, открывающие и закрывающие транзакцию в базу данных. Внутри этих тэгов можно описать любые модификации предыдущей версии базы. Поскольку любой блок описывает изменения относительно прошлой версии, то есть как-бы накапливает их от имеющихся, любую базу можно развернуть до некой временной отметки, используя только предложенное решение.

В процессе работы с документов система дополняет информацию в лог-файле методом автоотчета. После окончания работы, лог становится доступен для остальных клиник. Для синхронизации, сервер после получения лог-файла выполняет недостающие у него инструкции, приводя информацию о документе в актуальное состояние (рис. 5).

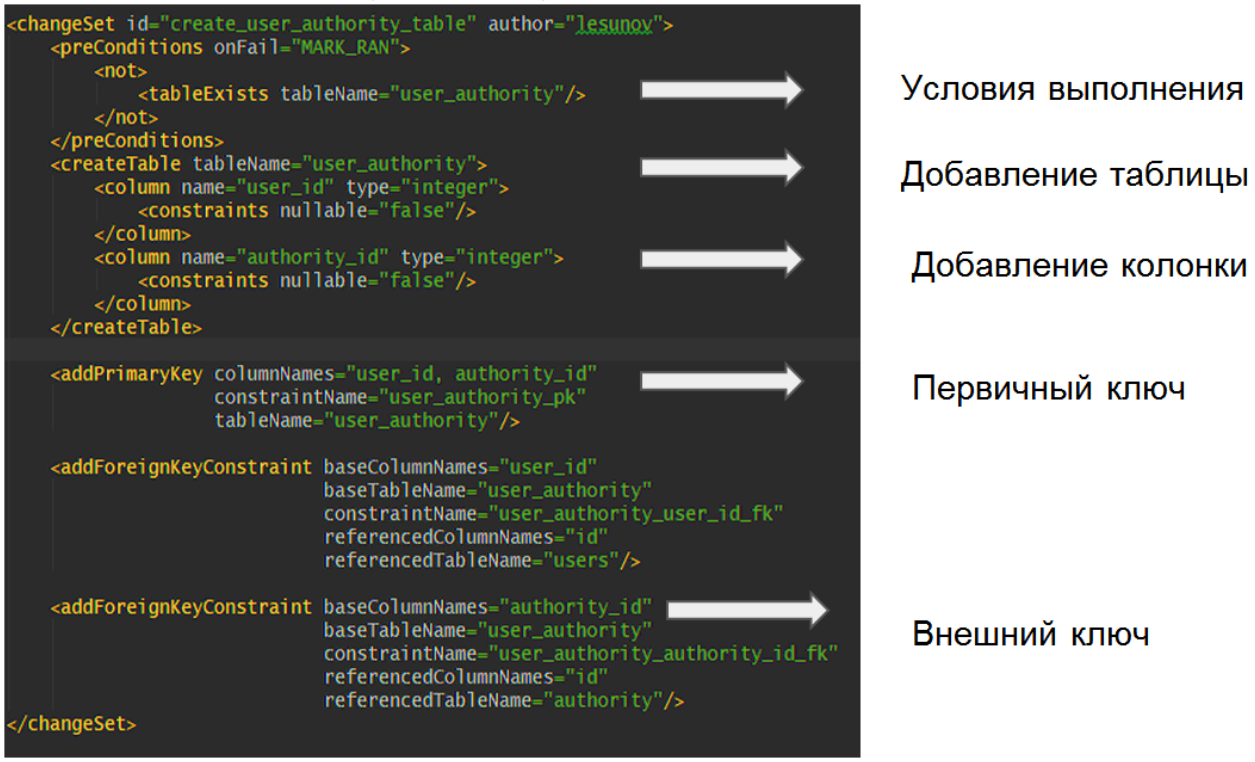


Рис. 5. Язык описания изменений

Заключение

Все изменения в базе данных хранятся в текстовых файлах (XML, YAML, JSON или SQL) и идентифицируются комбинацией тегов «id» и «author», а также именем самого файла. Список всех изменений хранится в каждой базе данных, с которой проводятся консультации по всем обновлениям базы данных, чтобы определить, какие новые изменения необходимо применять. В результате нет номера версии у базы данных, такой подход позволяет работать в средах с несколькими разработчиками и несколькими ветвями кода.

Литература

1. *Леонтьев, К. Б.* Комментарий к Федеральному закону «Об электронной цифровой подписи» / К. Б. Леонтьев. (постатейный). – М. : ООО «ТК Велби», 2013. – 6 с.

2. Ассоциация электронных торговых площадок. – Режим доступа: http://aetp.ru/ (дата обращения: 14.04.2021).

3. Рассылка о криптографии. – Режим доступа: http://www.metzdowd.com/ (дата обращения: 14.04.2021).

4. CoinTelegraph. – Режим доступа: :https://cointelegraph.com/explained/smart-contracts-explained/ (дата обращения: 14.04.2021).

Лесунов Павел Сергеевич – магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: pashales97@gmail.com

Анищев Тимур Владимирович – магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: anishchev.timur@ya.ru

Азарнова Татьяна Васильевна (научный руководитель) – д-р техн. наук, проф., заведующий кафедрой математических методов исследования операций Воронежского государственного университета. E-mail: ivdas92@mail.ru

УДК 004.021

РАСШИРЕНИЕ ИНТЕГРАЦИОННЫХ ВОЗМОЖНОСТЕЙ 1С С СЕМЕЙСТВОМ CAD-СИСТЕМ

Д. С. Литвинов

Воронежский государственный университет

Введение

«1С: Передприятие» – среда программ фирмы «1С», предназначенная для автоматизации деятельности предприятия. «1С: Предприятие» состоит из двух основных взаимосвязанных частей: прикладное решение (конфигурация) и технологическая платформа. Конфигураций на данный момент очень много (1С: Бухгалтерия, 1С: Управление торговлей, 1С: Зарплата и управление персоналом и т.д.), а технологическая платформа одна. В свою очередь, конфигурация устанавливается поверх платформы, после чего пользователи уже могут приступать к работе.

Гибкость платформы позволяет применять «1С: Предприятие» в разных областях: автоматизации производственных и торговых предприятий, бюджетных и финансовых организаций и т. д.; поддержки оперативного управления предприятием; автоматизации организационной и хозяйственной деятельности; расчета заработной платы и управления персоналом и других областях [1].

1. CAD-системы

CAD-система (**computer-aided design** – компьютерная поддержка проектирования) – программное обеспечение, которое автоматизирует труд инженера-конструктора и позволяет решать задачи проектирования изделий и оформления технической документации при помощи персонального компьютера.

Все CAD-системы, независимо от терминологии, предназначены для оптимизации работы инженерного состава предприятия. Если применять их правильно, уместно, они повышают производитбельность труда отдельных групп сотрудников. А это приводит к повышению общих показателей производительности персонала в целом.

CAD-комплексы, развернутые на предприятии, позволяют решить такие задачи:

- снизить трудоемкость отдельных операций и процессов, а значит уменьшить время и затраты на разработку, изготовление продукции;
- сократить время на подготовку проектов: с этими системами проектирование выводится на принципиально иной уровень;
- увеличить точность изготовления продукции без потерь в скорости (оперативность производства даже возрастает);
- снизить расходы, которые необходимы для содержания инженерного состава (что уменьшает себестоимость готового изделия);
- повысить качество проектирования: CAD-программы выводят его на новую технико-экономическую ступень;
- снизить расходы на моделирование образцов и проведение их испытаний [2].

2. Существующие решения

Существует готовое решение от фирмы 1С «1С:PDM Управление инженерными данными» (рис. 1). Это решение представляет из себя продукт, который предназначен для автоматизации конструкторско-технологической подготовки производства и поддерживает решение ряда задач по управлению информацией об изделии на протяжении его жизненного цикла, начиная от создания концепции изделия и НИОКР (научно-исследовательские и опытно-конструкторские работы), и заканчивая выводом изделия из эксплуатации.

Рис. 1. 1С:PDM Управление инженерными данными

Данное решение уже предусматривает наличие интеграции с двумя CAD-системами «КОМПАС-3D» и «SolidWorks». При этом перечень возможностей данной интеграции является ограниченным. Перечисленных возможностей достаточно для данной системы, т. к. назначением данного продукта является подготовка к производству, где CAD-система – это одна из нескольких применяемых систем.

Есть возможность получения следующей информации из CAD-систем:

- информация о характеристиках изделий;
- модель вторичного представления и его параметры.

Информация позволяет получить общее представление о физических свойствах изделия, его геометрических параметрах, стадии жизненного цикла.

Первое решение было выпущено ещё в 2005 году и продолжает поддерживаться и дорабатываться. Данное решение пользуется большим спросом у производственных предприятий и было внедрено более чем на 350 предприятиях различных отраслей и форм собственности, среди которых: ОАО «Борисовский завод агрегатов», ООО «Волгоградская машиностроительная компания «ВгТЗ», ЗАО «Завод им. Козицкого», АО «Каменск-Уральский литейный завод», ООО «Липецкий завод гусеничных тягачей», АО «Березниковский механический завод», АО «Заводоуковский машиностроительный завод», ЗАО «Волмаг», ООО «МИР», ООО «СТЕЛЛАЖИ МЕДВЕДЬ» и другие [3].

3. Предлагаемая схема

В предлагаемом мной решении по расширению интеграционных возможностей планируется предусмотреть функционал изменения состава изделий непосредственно из программы 1С и добавить возможность получения схем изделия, чертежей и т.д. Данное решение будет востребованным, так как не всем пользователям надо вносить правки в CAD-систему, а достаточно некоторых изолированных действий. Но для выполнения этих действий требуется отдельная лицензия для таких пользователей. Данное решение не потребует траты на это денежных средств. Предлагаемая мной разработка позволяет существенно расширить интеграционные возможности в сравнении с уже имеющимися продуктами, т.к. она направлено непосредственно на автоматизацию работы с CAD-системами. Она будет полезна при одновременном взаимодействии с несколькими системами, т.к. позволит вынести объемную часть работы непосредственно в 1С.

Принцип интеграции зависит от CAD-системы, возможны обращения через COM-объект, SOAP-протокол и т. д. **Component Object Model (COM)** – компонентно-ориентированная архитектура, которая позволяет создавать приложения и программные системы, построенные из совокупности компонентов, разработанных различными производителями и на разных платформах. COM является основой, на которой строятся более высокоуровневые приложения и сервисы [4]. **SOAP (Simple Object Access Protocol)** – простой протокол, основанный на XML, для обмена сообщениями в распределенных средах (WWW). Он предназначен для создания веб-сервисов и удаленного вызова методов [5]. Планируется реализовать интеграционный модель, у которого будет возможность интеграции с несколькими системами, одна из которых «Компас-3D».

Со стороны 1С для постоянной актуализации информации будет использоваться механизм регламентного задания, который будет автоматически запускаться с определенной периодичностью. Главной целью этого регламентного задания будет получение актуальных данных из CAD-системы. Регламентное задание будет обращаться по API к системе и загружать данные, переданные из 1С, а затем делать запрос на выгрузку данных в каталог обмена. **API (application programming interface)** – интерфейс прикладного программирования, набор методов, классов, библиотек, функций, обеспечивающих взаимодействие программ между собой [6]. Затем это же регламентное задание будет забирать выгруженные файлы, считывать их и загружать в базу данных 1С, в предварительно созданные для этого объекты метаданных (рис. 2). Главным отличием от существующих решений будет расширение возможностей и функционала интеграций с CAD-системами.

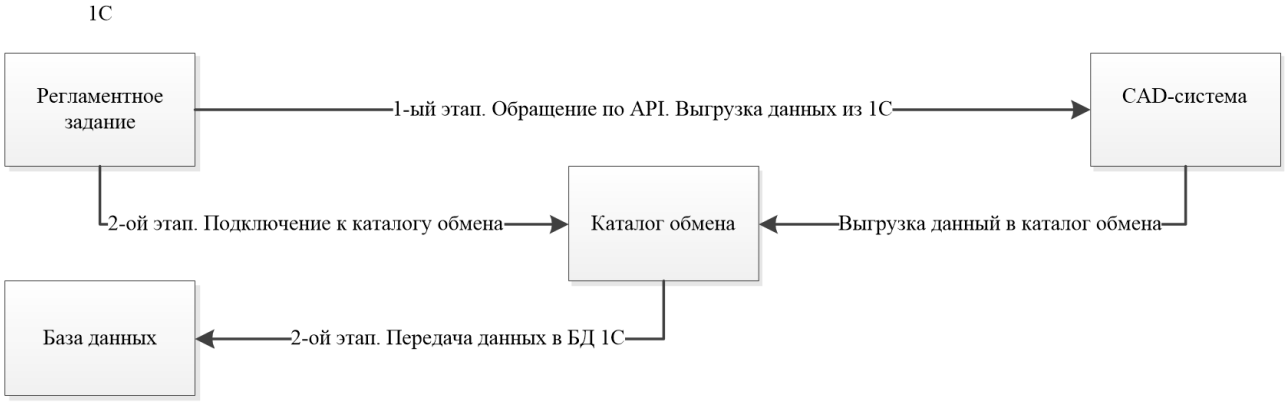


Рис. 2. Схема интеграции

Заключение

Благодаря широким возможностям 1С в области интеграций с внешними системами, в разрабатываемом решении есть возможность взаимодействия с CAD-системами различными способами, что позволяет существенно расширить интеграционные возможности. Данная разработка помогает сократить расходы на лицензии для приложений из семейства CAD-систем, т.к. объемная часть CAD функционала будет доступна из 1С. Данная разработка направлена непосредственно на автоматизацию работы с CAD-системами. Она будет полезна при одновременном взаимодействии с несколькими системами, т.к. позволит вынести объемную часть работы непосредственно в 1С. Все вышеперечисленные достоинства делают данное решение актуальным и востребованным.

Литература

1. *Новикова, Т. И.* Особенности и преимущества платформы «1С: Предприятие» / Т.И. Новикова, Ю.А. Толстикова // Актуальные проблемы авиации и космонавтики. – 2015. – Т. 1. – С. 592–593.

2. CAD/CAM. – Режим досутпа: http://planetacam.ru/college/learn/12-2/ (дата обращения: 05.04.2021).

3. 1С: PDM Управление инженерными данными. – Режим досутпа: https://solutions.1c.ru/catalog/plm/features (дата обращения: 01.04.2021).

4. *Близнюк, А. В.* Создание и применение компонентов com / А. В. Близнюк // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. – 2010. – № 1. – С. 117–118.

5. *Заленский, Д. А.* Разработка универсальной модульной системы удаленной обработки геоданных, основанной на технологии soap (Simple Object Access Protocol) / Д. А. Заленский // Киберленинка. – 2006. – № 4. – С. 31–35.

6. API. – Режим доступа: https://promo.ingate.ru/seo-wikipedia/api/ (дата обращения: 18.04.2021).

Литвинов Даниил Сергеевич – магистрант 1-го года обучения кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: keks2013@mail.ru

Сафронов В.В. (научный руководитель) – канд. техн. наук, доц., доцент кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: vitolik@bk.ru

УДК 004.023

РЕШЕНИЕ ЗАДАЧИ МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ С ИСПОЛЬЗОВАНИЕМ МУРАВЬИНОГО АЛГОРИТМА

А. А. Ломова

Воронежский государственный университет

Введение

В современном мире спрос на транспортно-логистические услуги повышается ежегодно. Еще в 1990-х годах, когда транспортный бизнес только начинал развиваться, всего несколько транспортных компаний могли принимать заказы на грузовые перевозки, в то время как сейчас функционируют крупные логистические компании, охватывающие 90 % всех перевозимых грузов [5]. Наблюдается стремительно растущий спрос на товары, вследствие чего и увеличивается объем перевозок. Транспортные компании стремятся получить конкурентное преимущество перед другими компаниями, в частности с помощью снижения издержек, при это увеличивая прибыль. Оптимизация затрат может быть достигнута в том числе и за счет построения эффективных маршрутов для транспортных средств.

1. Описание задачи маршрутизации транспортных средств

Задача маршрутизации транспортных средств (ТС) представляет собой целый класс задач комбинаторной оптимизации, в которых набор маршрутов для парка автомобилей, расположенных в одном или нескольких депо, должен быть определен для нескольких географически отдаленных городов или покупателей. Целью решения задач маршрутизации является оптимизация выполнения серии запросов клиентов. Данная задача является обобщением задач коммивояжера и задачи о рюкзаке, а, следовательно, принадлежит к классу NP-трудных задач – сложность экспоненциально возрастает с увеличением размера входных данных. Очевидно, что точные подходы решения целесообразно использовать только при малоразмерных задачах, в реальных же условиях не только увеличивается размерность, но и накладывается множество ограничений.

Для решения задач маршрутизации транспорта чаще всего используются эвристические и метаэвристические алгоритмы. Они характеризуются нахождением оптимального или близкого к оптимальному решения за короткий промежуток времени. Эвристический алгоритм – это алгоритм, основанный на некотором правиле, не полностью математически обоснованный, но при этом практически полезный. То есть на выходе мы получаем допустимое решение, которое «достаточно приемлемо» в рамках «достаточно хорошего» вычислительного времени. Метаэвристики объединяют основные эвристические методы в рамках алгоритмических схем более высокого уровня [8]. Удобство метаэвристик заключается в том, что они описываются на абстрактном уровне, а значит могут быть адаптированы для большинства задач оптимизации, обеспечивая лучший компромисс между качеством решения и вычислительным временем.

1.2. Природные вычисления

На сегодняшний день активно развивается такое научное направление как «Природные вычисления», включающее в себя математические методы, которые опираются на заимствованные у природы процессы, явления, модели поведения. Биологические механизмы обеспе-

чивают реальную эффективность, адаптацию флоры и фауны к окружающей среде на протяжении миллионов лет. Популярность такого рода вычислений связана с созданием новых парадигм вычислений, искусственных биологических систем, с необходимость осуществления параллельных вычислений. Среди разделов природных вычислений можно выделить такие, как генетические алгоритмы, клеточные автоматы, эволюционное программирование, а также муравьиные алгоритмы [9].

Муравьиные алгоритмы применяются к различным транспортным задачам и их модификациям. Исследования данной области начались в середине 90-х годов XX в., автором идеи является доктор наук Марко Дориго, который предложил использовать моделирование поведения колонии муравьев. Муравьи относятся к группе социальных насекомых – это насекомые, отличающиеся от других общественным образом жизни. Сохранение всей сложной структуры муравьиной колонии, связей особей между собой обусловлено химической коммуникацией – муравьи общаются с помощью феромонов. Большую часть жизни они проводят в контакте с землей, поэтому поверхность почвы является хорошим местом, чтобы оставлять след феромона, который может ощущаться другими муравьями с помощью длинных и тонких усиков. В частности, муравьи используют феромоны для прокладки маршрутов: нашедший пищу отмечает путь до муравейника, по которому передвигаются другие муравьи. Те, в свою очередь, проходя по этому маршруту, отмечают его своими феромонами. Когда путь перестает быть актуальным (источник пищи исчерпан), то насекомые перестают помечать его, вследствие чего концентрация феромонов постепенно уменьшается.

1.3. Постановка задачи

Задачу маршрутизации транспортных средств можно сформулировать следующим образом: имеется несколько парков (депо) транспортных средств одинаковой грузоподъемности, а также множество пунктов доставки с запросами на доставку. Необходимо распределить маршруты с учетом минимальных издержек так, чтобы каждый пункт доставки был посещен только один раз, а маршрут каждого ТС начинался и заканчивался в депо.

1.4. Математическая модель

В простом муравьином алгоритме, использованном в данной статье, рассматривается депо, состоящее из n транспортных средств.

Пусть $G(V,U)$ – взвешенный граф, где

- $V = \{V_0, v_1 ..., v_n\}$ – множество вершин, причем V_0 – множество начальных вершин, т. е. в которых расположены депо, $V' = \{V / V_0\}$ – множество вершин, где расположены пункты доставки.
- $U = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ – множество ребер.
- C – матрица расстояний, где c_{ij} – расстояние между вершинами v_i и v_j
- d – вектор запросов пунктов доставки.
- R_i – маршрут i-го транспортного средства.
- m – количество идентичных ТС, причем каждому ТС назначается один маршрут.
- a – вектор принадлежности ТС к депо.
- a_i – депо, к которому относится i-е транспортное средство.

В случае $c_{ij} = c_{ji}$ множество ребер U заменяют множеством граней $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$.

С каждой вершиной $v_i \in V'$ ассоциировано некоторое количество товаров, которое должно быть доставлено ТС. Задача сводится к минимизации суммарной длины маршрутов – расхода топлива, числа задействованных транспортов. Решением для данной постановки задачи являются маршрутные листы для каждого транспортного средства.

2. Муравьиный алгоритм

Опираясь на вышесказанное, основной идеей данного алгоритма является моделирование поведения муравьев. Выделим ключевые моменты:

1) Каждый муравей способен улавливать след феромона – это стимулирует его желание пройти по данному ребру.

2) Каждый муравей способен «хранить в памяти» список вершин, которые ему необходимо посетить.

2) Пройденное ребро муравей помечает своим феромоном, чем увеличивает его концентрацию на данном ребре.

3) Равномерно по всему множеству вершин происходит испарение феромона, что ограничивает уровень концентрации феромонов, а также обеспечивает обратную связь. Путь, на который было потрачено меньше времени, менее подвержен испарению, следовательно, концентрация феромонов на оптимальном маршруте будет дольше сохраняться.

Построение маршрута начинается с некоторой начальной вершины, и на каждом шаге к ней добавляются другие вершины, руководствуясь некоторым вероятностным правилом:

$$\begin{cases} P_{ij,k}(t) = \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{l \in J_{i,k}}^{n} [\tau_{il}(t)]^{\alpha}[\eta_{il}(t)]^{\beta}}, j \in J_{i,k} \\ P_{ij,k}(t) = 0, j \notin J_{i,k} \end{cases},$$

где

• $\eta_{ij} = \dfrac{1}{c_{ij}}$ – величина, обратная весу ребра U_{ij}, так называемое «зрение» муравья. Чем ближе находится вершина, тем лучше видимость и тем больше муравей стремится попасть в неё.

• $J_{i,k}$ – список вершин, которые необходимо посетить k-му муравью, находящемуся в городе i.

• α, β – параметры, задающие веса следа феромона, коэффициенты эвристики. При $\alpha = 0$ муравей стремится выбирать кратчайшее ребро, при $\beta = 0$ – ребро с наибольшим количеством феромона.

Вершины характеризуются уровнем феромона и эвристической информацией. Таким образом, для k-го муравья вероятность $P_{ij,k}$ перехода из вершины i в вершину j зависит от двух величин:

• привлекательности перехода («зрением» муравья) η_{ij} из вершины i в вершину j;

• уровня феромона τ_{ij} из вершины i в вершину j.

Количество феромона на ребре описывается соотношением:

$$\Delta\tau_{ij,k}(t) = \begin{cases} \dfrac{Q}{L_k(t)}, (i,j) \in R_k(t) \\ 0, \quad (i,j) \notin R_k(t) \end{cases},$$

где Q – параметр, имеющий значение порядка длины оптимального пути, $L_k(t)$ – длина маршрута $R_k(t)$.

Испарение феромона описывается следующим выражением:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij,k}(t),$$

где n – количество ТС, $p(0 < p < 1)$ – коэффициент испарения.

Временная сложность алгоритма зависит от количества итераций t, вершин графа m и числа муравьев n: $O(tmn^2)$. Существует множество модификаций алгоритма, позволяющих

улучшить временную сложность. Например, для ускорения сходимости вводят «элитных муравьев» – муравьев, чьи маршруты лучше остальных.

Качество решения во многом зависит от начального расположения. В простом муравьином алгоритме все муравьи первоначально расположены в разных вершинах, причем в каждой по одному (стратегия «одеяло»). В таком случае временная сложность алгоритма $O(tn^3)$. Существуют и другие варианты размещения: «дробовик» – случайно распределение муравьев, причем численность муравьев в разных вершинах может быть разной, а также в частности для задачи маршрутизации с одним депо – «фокусировка» – все муравьи находятся в одной вершине. В большинстве случаев наилучшее решение находится колонией, размерность которой значительно превышает число вершин [3].

Заключение

В данной статье рассмотрено применение муравьиного алгоритма для задачи маршрутизации транспортных средств. Приведены описание и математическая постановка задачи. Описаны цели решения и накладываемые ограничения. Оптимизация транспортных услуг за счет построения эффективных маршрутов в условиях рыночной экономики позволит уменьшить не только транспортные затраты, но и себестоимость продукции.

Литература

1. *Агеев, Е. В.* Обзор природных вычислений: основные направления и тенденции / Е. В. Агеев, Е. Н. Бендерская // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – 2014. – № 2 (193). – С. 9–22.

2. *Гончарова, Ю. А.* Задачи маршрутизации при транспортировке: обзор моделей, методов и алгоритмов. Часть 1 / Ю. А. Гончарова, Р. С. Валеев, А. Ф. Валеева // Логистика и управление цепями поставок. – 2019. – № 4 (93). – С. 74–88.

3. Кажаров, А. А. Муравьиные алгоритмы для решения транспортных задач / А. А. Кажаров, В. М. Курейчик // Известия Российской академии наук. Теория и системы управления. – 2010. – № 1. – С. 32–45.

4. *Кирсанов, М. В.* Графы в Maple. Задачи, алгоритмы, программы: учебное пособие / М. В. Кирсанов. – Москва : Физмалит, 2007. – 168 с.

5. *Ключников, М. В.* Рынок автотранспортных перевозок: перемены к лучшему / М. В. Ключников // Региональная экономика: теория и практика. – 2004. – № 8. – С. 39–42.

6. *Токарева, И. О.* Муравьиный алгоритм для полной и неполной постановок задачи / О. И. Токарева, А. Б. Гончарова, Е. И. Сергеева // Естественный и математические науки в современном мире. – 2017. – № 1 (48). – С. 30–35.

7. *Штобва, С. Д.* Муравьиные алгоритмы / С. Д. Штобва // Exponenta Pro. Математика в приложениях. – 2003. – № 3. – С. 70–75.

8. *Щербина, О. А.* Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор) / О. А. Щербина // Таврический вестник информатики и математики. – 2014. – № 1 (24). – С. 56–72.

9. *Kari, L.* The Many Facets of Natural Computing / L. Kari, G. Rozenberg // Communications of the ACM. – 2008. – Vol. 51, Issue 10. – P. 72–83.

Ломова Анастасия Алексеевна – студент 3-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: anastlomova14@gmail.com

Авсеева Ольга Владимировна (научный руководитель) – канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: olga-avseeva@mail.ru

УДК 004.04

N-ГРАММА ДЛЯ ПРОЕКТИРОВАНИЯ АРХИТЕКТУРЫ ЧАТ-БОТА С ОБРАБОТКОЙ ЕСТЕСТВЕННОГО ЯЗЫКА

А. А. Мазуров

Воронежский государственный университет

Введение

Одним из популярных на сегодняшний день GUI является чат-бот или же письменный диалог между пользователем и сервисом [4].

Основными сложностями в данном UI является распознавание фразы человека (например, в случае допущения пользователем грамматической ошибки в слове) и выявление сути с последующим корректным ответом. На сегодняшний день существует достаточно алгоритмов для решения этих задач, отличающихся по времени обработки и сложности.

Например, одним из таких решений является бот, построенный на машинном обучении и нейронных связях. Его неоспоримыми преимуществами является гибкость в диалоге с пользователем, а также вариативность задач, которые бот может решать. Но главным камнем преткновения является сложность разработки и обучения.

Альтернативой является решение, которое основывается на N-грамме и словаре. Главным преимуществом является скорость разработки и низкая стоимость по мощностям ЭВМ. Основная цель такого решения – создание сервисного чат бота с ограниченным набором фраз-команд.

1. Анализ задачи

1.1. N-грамма

N-грамма – последовательность из n элементов. С семантической точки зрения в процессе обработки предложения для данной задачи – это последовательность слов. Каждое слово или устойчивое выражение это токен, последовательность слов это N-грамма токенов. Например, предложение «Открой сайт помощи абитуриентам» разобьется следующим способом (рис. 1). То есть, сообщение пользователя разбивается на отдельные слова, разбиваясь по пробелам и знакам препинания.

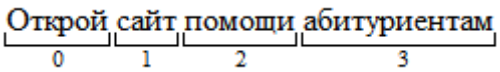


Рис. 1. N-грамма

Далее N-грамма передается в компонент распознавания токенов, где проверяется каждый токен на наличие в нем орфографических и грамматических ошибок.

1.2. Расстояние Левенштейна

Расстояние Левенштейна, или расстояние редактирования между двумя строками – это минимальное количество операций вставки, удаления и замены символа на другой, необходимых для превращения одной строки в другую [2].

Пусть S_1 и S_2 – две строки длиной M и N соответственно. Тогда расстояние Левенштейна можно рассчитать по следующей рекуррентной формуле (первый элемент строки имеет

номер 1): $d(S_1, S_2) = D(M, N)$, где справедливы следующие условия (1), где $m(a,b) = 0$, если $a = b$, или 1 – в противном случае.

$$D(i,j) = \begin{cases} 0, \ i = 0, \ j = 0 \\ i, \ j = 0, \ i > 0 \\ j, \ i = 0, \ j > 0 \\ \min\{D(i, j-1)+1, D(i-1, j)+1, D(i-1, j-1)+m(S_1[i], S_2[j]), \ j > 0, i > 0\} \end{cases} \quad (1)$$

Здесь шаг по i символизирует удаление из первой строки, по j – вставку в первую строку, а шаг по обоим индексам символизирует замену символа или отсутствие изменений. Справедливы следующие утверждения расстояние Левенштейна (2):

$$\begin{aligned} & d(S_1, S_2) \geq \| S_1 | - | S_2 \| \\ & d(S_1, S_2) \leq \max(| S_1 |, | S_2 |) \\ & d(S_1, S_2) = 0 \Leftrightarrow S_1 = S_2 \end{aligned} \quad (2)$$

1.3. Распознавание и исправление ошибок

Во время прохода по N-грамме, выполняется поиск каждого слова в словаре, если такого слова не найдено, то осуществляется поиск слова. В поисках оптимального алгоритма по исправлению ошибок с помощью словаря было проанализировано несколько алгоритмов.

1.3.1. Наивный подход

Заключается в подборе слова из словаря по расстоянию Левенштейна. Для замены выбирается слово с минимальным расстоянием, так же вводятся дополнительные условия для исключения нескольких вариантов. Главный минус заключается в необходимости обработки огромного массива данных, который будет присутствовать в задаче [2].

1.3.2. Стратегия промаха

Если слово не найдено в словаре, то генерируются все возможные комбинации слова с заданным расстоянием редактирования и выполняется их поиск в словаре. Если эти шаги приводят к слову, содержащемуся в словаре, то оценивается расстояние до исходного слова. Чтобы измерить близость слов, используется расстояние Левенштейна. Из числа найденных слов для замены выбирается слово с наименьшим расстоянием Левенштейна [2].

Метод весьма ресурсоемок. Так, для слова длины n, размера алфавита a, расстояния редактирования $d = 1$, будут выполнены n удалений букв, $n-1$ перестановок, $a * n$ изменений и $a*(n+1)$ вставок. В общей сложности число получаемых модификаций исходного слова равно $2n + 2an + a - 1$.

Этот способ хоть и лучше наивного подхода, однако все же весьма дорогостоящий по времени поиска. Так для $n = 9, \ a = 36, \ d = 2$ получается 114 324 комбинации.

Достоинство алгоритма: использование расстояния Левенштейна, что повышает точность ответов.

Основные недостатки: большое количество дочерних операций, продолжительное время работы, не учитывается контекст.

1.3.3. Стратегия для решения задачи

За основу данного алгоритма будет взята стратегия промаха. Ограничив операции лишь удалением, необходимо составить всевозможные N-граммы для слова, удаляя случайные $N = kr$ (N округляется до целого в большую сторону) букв из слова, и дробя полученную стро-

ку по индексам удаления, где k – длина строки. Получаем, что число модификаций исходного слова считается по формуле сочетания $x = C_k^N$, что сводит к значительно меньшему количеству модификаций, полученных в стратегии промаха.

Далее необходимо сделать всевозможные выборки из словаря с пересечением по каждому элементу каждой N-граммы, объединив выборки будет получен новый словарь, со схожими с искомым словами.

Теперь необходимо посчитать расстояние Левенштейна начального слова с каждым, из нового словаря, и выбрать с наименьшим значением. Если такое слово не единственное, то выбрать случайное.

Необходимо рассмотреть число r из формулы, оно подбирается опытным путем, в зависимости от среднего количества ошибок, которые допускает пользователь и мощностей машины, на которой будет запускаться алгоритм. Соответственно чем меньше r, тем больше N, что означает большее число разбиений для слова и большее число всевозможных модификаций.

1.4. Лемматизация и стемминг текста

Обычно тексты содержат разные грамматические формы одного и того же слова, а также могут встречаться однокоренные слова. Лемматизация и стемминг преследуют цель привести все встречающиеся словоформы к одной, нормальной словарной форме [3].

Например, из (рис.1) слово под индексом 2 «помощи» является словоформой слова «помощь».

Стемминг – это грубый эвристический процесс, который отрезает «лишнее» от корня слов, часто это приводит к потере словообразовательных суффиксов.

Лемматизация – это более тонкий процесс, который использует словарь и морфологический анализ, чтобы в итоге привести слово к его канонической форме – лемме [3].

Отличие в том, что стеммер действует без знания контекста и, соответственно, не понимает разницу между словами, которые имеют разный смысл в зависимости от части речи. Однако у стеммеров есть и свои преимущества: их проще и быстрее реализовать в проекте, а так же у них ниже время обработки. Плюс, более низкая «аккуратность» может не иметь значения в некоторых случаях.

Для задачи был выбран способ лемматизации, чтобы сократить количество поиска по словарю. А также по причине использования анализа текста в виде сервисного UI, нужно более точное осознание того, что сказал пользователь, в связи с сильной вариативностью русского языка и словообразования в нем.

1.5. Стоп-слова и выделение смысла

Следующим этапом необходимо определить смысл фразы пользователя, чтобы выделить ключевые слова. В словаре на каждом нормированном слове есть тэги, позволяющие построить ответ на базе уже известных фраз-команд. Для этого, перебирая каждый элемент полученной N-граммы, выполняется поиск по словарю, и если у слова нет тэга, отсылающего его к командной фразе, то оно помечается соответствующим флагом или удаляется из последовательности.

Удалив шум, необходимо получить дальнейшую команду. В случае задачи, пользователь хочет получить ссылку на сайт (рис. 1). Необходимо сделать выборку среди фраз-команд по тэгам, представленным на оставшихся словах. Например, слова «помощь» и «абитуриент» будут иметь тэги, указывающие на url страницы сайта. Но также слово «помощь» имеет тэг, который указывает на формирование ответа пользователю со списком команд, в этом случае

необходимо учитывать иерархию тэгов и формировать ответ, не только, по отдельным словам, но и по целым фразам, проверяя совпадение, полученной нормированной N-граммы и фразы-команды.

Заключение

Предложенная модель, основанная на N-граммах, помогает справиться с определенным кругом задач анализа языка по предварительно сгенерированному словарю слов, а именно, автоматически анализировать запросы клиентов и генерировать максимально правильные и возможные ответы на эти запросы. При этом модель не требует больших ресурсов в разработке, а также высокой мощности машины, на которой будет запускаться.

В ходе решения задачи был придуман и протестирован алгоритм поиска ошибок и автоматического исправления слова, не требующий нейронных сетей, с относительно высокой скоростью работы.

Литература

1. *Джанарсанам, С.* Разработка чат-ботов и разговорных интерфейсов / С. Джанарсанам. – Москва : ДМК-Пресс, 2019. – 340 с.

2. Методы автоматического поиска и исправления ошибок в предложении. – Режим доступа: http://www.100byte.ru/stdntswrks/spellCh/spellCh.html

3. Основы Natural Language Processing для текста. – Режим доступа: https://habr.com/ru/company/Voximplant/blog/446738/

4. *Болотова, С. Ю.* Использование продукционной модели знаний для решения задачи генерации подмножеств естественного языка / С. Ю. Болотова, К. В. Сиволапов // ИНФОРМАТИКА: ПРОБЛЕМЫ, МЕТОДЫ, ТЕХНОЛОГИИ Материалы XX Международной научно-методической конференции. – 2020. – С. 25–30.

Мазуров Алексей Алексеевич – студент 4-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: voidzarik@gmail.com

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: bolotova.svetlana@gmail.com

УДК 004.04; 004.6

ОСОБЕННОСТИ ВИЗУАЛИЗАЦИИ ДАННЫХ В СИТУЦИОННЫХ ЦЕНТРАХ СУБЪЕКТОВ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ю. А. Максименко, В. В. Ухлова

Воронежский государственный университет

Введение

Одной из задач ситуационных центров (СЦ), функционирующих в субъектах Российской федерации, является обеспечение мониторинга региона в сфере безопасности. В свою очередь, это позволит оперативно и своевременно реагировать на возникающие угрозы, оптимально распределять критичные ресурсы (реанимобили, отряды МЧС и другие). Особое место в плане безопасности любого региона занимает дорожно-транспортная ситуация. В частности, особую важность имеет мониторинг дорожно-транспортных происшествий (ДТП). Это связано с тем, что данный тип аварий занимает лидирующее положение в списке чрезвычайных ситуаций (ЧС) регионов по количеству случаев, а также числу погибших и пострадавших. Обеспечение мониторинга позволит оценить текущую ситуацию на дорогах региона и выявить наиболее загруженные и опасные участки, позволив, тем самым, оптимизировать движение и сократить число аварий и жертв. В данной работе рассматривается задача мониторинга дорожно-транспортной ситуации, решаемая в ситуационных центрах РФ.

1. Постановка задачи

В соответствии с ФЗ от 10.12.1995 № 196-ФЗ (ред. от 08.12.2020) «О безопасности дорожного движения» и правилами дорожного движения (ПДД) под дорожно-транспортным происшествием понимается событие, возникшее в процессе движения по дороге транспортного средства (ТС) и с его участием, при котором погибли или ранены люди, повреждены транспортные средства, сооружения, грузы либо причинен иной материальный ущерб [1]. Таким образом, мониторинг дорожно-транспортной ситуации относительно безопасности заключается в получении, обработке и отображении информации о вышеуказанных событиях на дороге.

Основными источниками информации о ДТП в субъектах РФ являются:

1) на государственном уровне официальный сайт Государственной Инспекции Безопасности Дорожного Движения (ГИБДД) и государственная система экстренного реагирования ЭРА-ГЛОНАСС;

2) на муниципальном уровне информационные системы единой дежурно-диспетчерской службы (ЕДДС) и дежурно-диспетчерской службы (ДДС), «Система-112», единый центр оперативного реагирования (ЕЦОР).

Помимо этого, в каждом регионе источниками официальной информации о ДТП можно считать региональные навигационные информационные системы и городские системы видеонаблюдения, которые контролируют движение муниципальных транспортных средств (машины скорой помощи, школьные автобусы, дорожную технику, ведомственный транспорт и пассажирские перевозки) и предоставляют потоковые видео с камер видеофиксации на дорогах региона. Из неофициальных источников следует выделить группы (сервисы) в телекоммуникационной сети Интернет, отображающие ситуацию на дорогах («пробки», ДТП, аварии коммунальных служб). Немаловажным остается и тот факт, что не все ДТП фиксируются, т. к. не все приводят к неблагоприятным последствиям на дороге.

Наиболее удобным способом мониторинга ДТП является визуализация информации. Такое решение позволяет главам регионов принимать эффективные управленческие решения с минимальными временными затратами, связанными с анализом данных и интерпретацией результатов. Задача исследования формулируется следующим образом: оценить возможности мониторинга дорожно-транспортной ситуации в СЦ субъектов РФ, выбрать оптимальные с позиции реализации средства визуализации и разработать сценарий визуализации имеющихся данных.

2. Возможности визуализации информации в СЦ

В рамках данной работы речь будет идти только о визуализации, которую можно реализовать на основании официальной информации. Основными источниками информации о ДТП на муниципальном уровне для СЦ являются информационные системы ГИБДД, «Система-112» и ЕЦОР. Каждая из этих систем предоставляет возможность получения СЦ: координат аварий, времени наступления события, данных о категории ДТП, описание происшествия, а также данных о транспортных средствах.

Для отображения статистических данных СЦ использует различные способы визуализации, наиболее востребованными являются: карты, диаграммы, гистограммы и графики. Видео с камер видеофиксации, также, транслируется в СЦ.

Средства визуализации СЦ позволяют:

1) создавать тепловые карты, карты с условной и градиентной раскраской;

2) выводить диаграммы, гистограммы и графики для построения сравнительной аналитики и отображения сосредоточений и выбросов в данных;

3) строить временные ряды;

4) объединять данные из разных источников на одной карте и выводить информацию с разделением по слоям;

5) осуществлять агрегацию данных.

3. Алгоритм визуализации данных

Для решения задачи мониторинга ДТП, рассмотренными выше средствами, алгоритм визуализации представим следующими пятью шагами.

Шаг 1. Изучение источника данных.

В качестве источников данных выступают ведомственные системы, интеграции с которыми регламентируются указами и распоряжениями представителей органов власти, как на федеральном, так и на региональном уровне. На данном этапе происходит описание и анализ модели данных (формат, состав, объем), структуры хранения, способа доступа и передачи данных.

Шаг 2. Определение инструментов визуализации.

После того, как было получено представление о запрашиваемых данных, необходимо выбрать способ конечной визуализации. Данный выбор позволяет понять соответствует ли формат запрашиваемых данных требованиями BI-решения СЦ. В случае несоответствия необходимо разработать алгоритм приведения данных к виду, отображаемому в СЦ.

Шаг 3. Получение запрашиваемых данных.

Считается, что на этом шаге данные собраны и ставится задача только их получить. Если источник данных единственный, то получение осуществляется путем выгрузки или с помощью API. Если источников несколько, то требуется предварительная интеграция данных. Если исходные данные являются статичными и в рассматриваемой ИС существует возможность выгрузки данных в базу данных (БД), то осуществляется получение экземпляра БД посред-

ством файлового обмена. Для потоковых данных из источника, который имеет собственное API, целесообразно воспользоваться данным способом передачи данных.

Шаг 4. Преобразование данных.

В зависимости от интеграции, данный этап может быть реализован двумя способами [2]:

1) данные, приходящие из сторонней ИС, заранее не преобразуются и сохраняются в БД СЦ в исходном виде, после чего уже на стороне СЦ с помощью ETL-инструментов приводятся к нужному формату и попадают в БД, откуда используются BI решением;

2) данные, приходящие из сторонней ИС, сразу приводятся к формату необходимому для представления в BI.

Шаг 5. Визуализация данных.

Данный этап заключается в создании и настройке виджетов с последующим вынесением их на дашборд.

Следует понимать, что алгоритм визуализации статических и потоковых данных будет несколько отличаться. Это касается шагов 3 и 4.

Рассмотренный алгоритм визуализации данных представлен на рис. 1 [3, 4]. Он является традиционным, поэтому будем называть его «классическим».

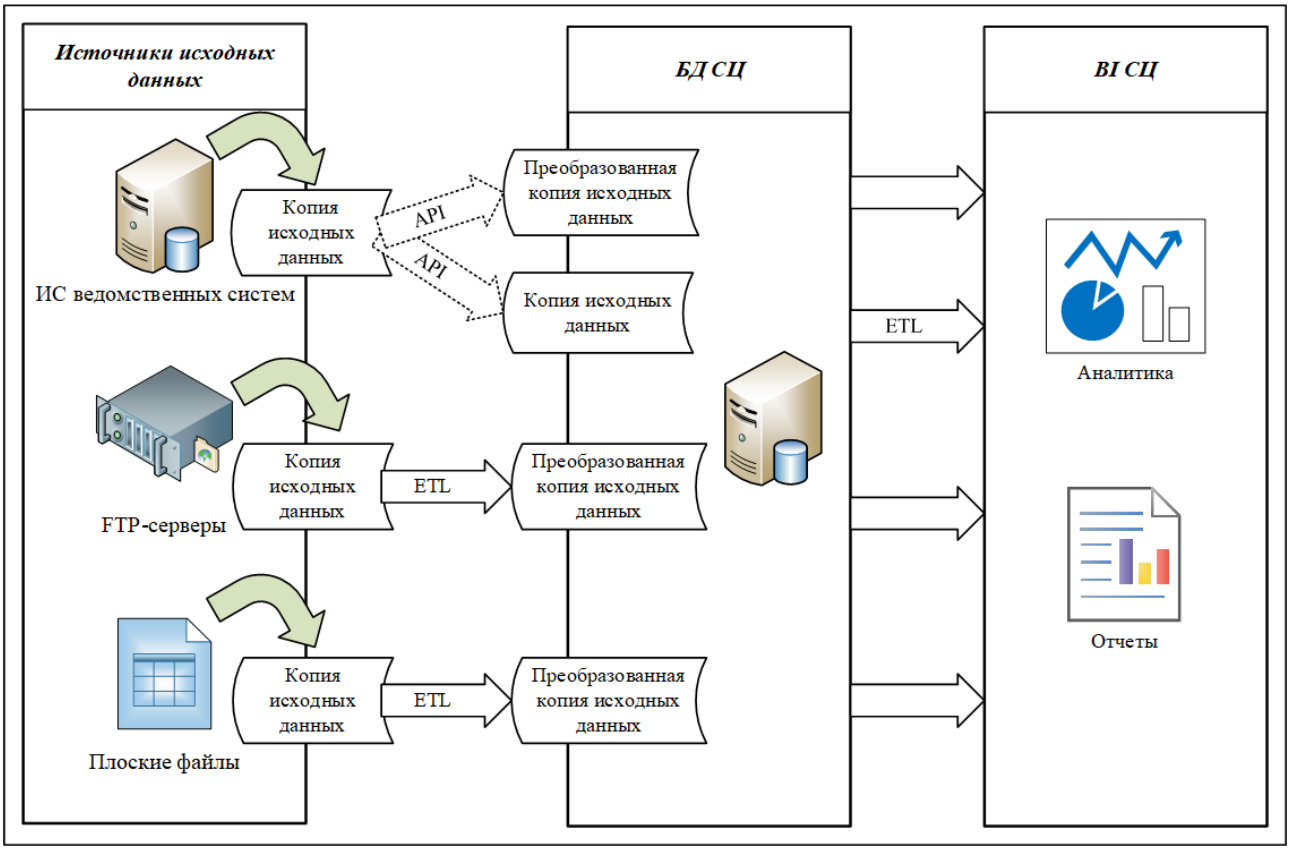


Рис. 1. «Классический» алгоритм визуализации

4. Проблемы визуализации данных по «классическому» алгоритму

Алгоритм, рассмотренный в п. 2 отображает текущий порядок шагов необходимый для визуализации данных. Однако такое решение нельзя назвать оптимальным с точки зрения удовлетворения аналитических потребностей конечных пользователей. Для достижения более оперативной и удобной работы необходима архитектура с дополнительным промежуточным уровнем в виде витрин данных, т. к. интеграция и согласование данных из различных источников в реальном времени занимает слишком много времени [5]. Скорость предостав-

ления данных в систему визуализации (BI) осложняется медленным выполнением операций записи и чтения данных. Поскольку уровень BI-решений предполагает эффективное хранение данных для обеспечения быстрого доступа и анализа – использование витрин данных является распространенной практикой при внедрении систем такого класса. Последовательность шагов с добавлением вышеописанного уровня представлена на рис. 2.

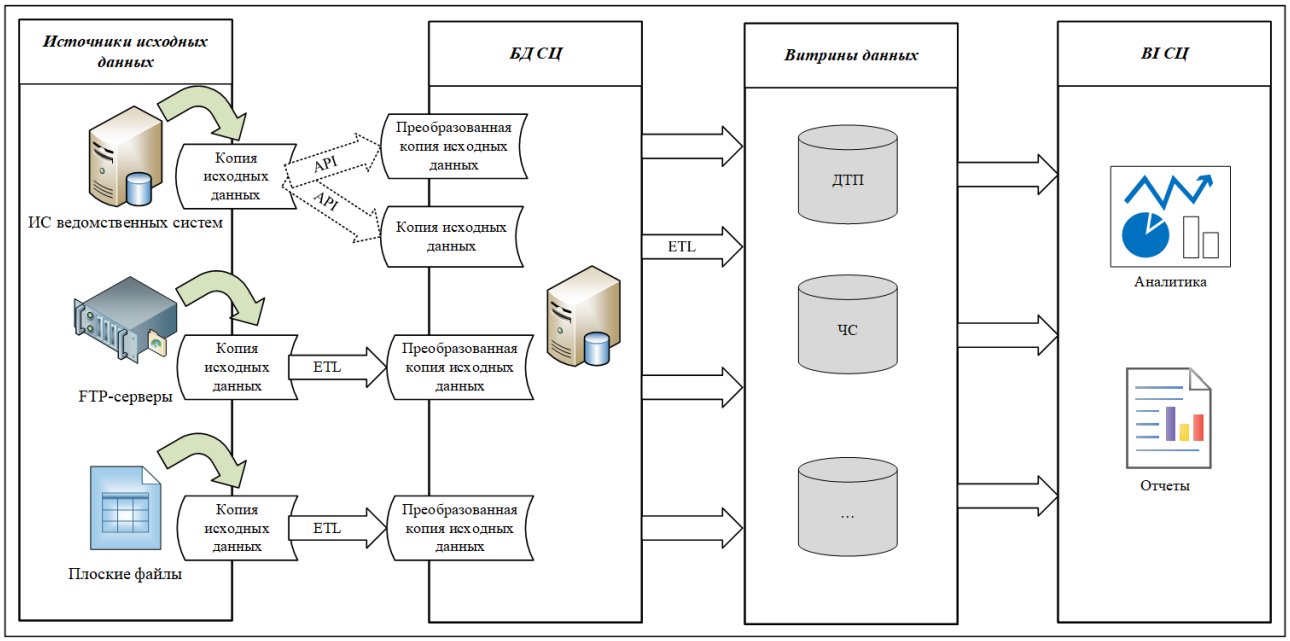


Рис. 2. Усовершенствованный алгоритм визуализации данных

Место данного шага между шагами 4 и 5 «классического» алгоритма. Таким образом, данное решение позволит удовлетворять потребности конечных пользователей быстрее за счет разбиения информации по тематическим группам. Также, такое решение имеет более низкую стоимость по сравнению с реализацией полноценного хранилища данных (ХД) и обеспечивает более высокий уровень безопасности и контроля над данными за счет партицирования. Витрины данных могу содержать данные не только в разрезе ДТП, но и других направлений необходимых для решения аналитических задач СЦ.

Заключение

Задачи визуализации являются достаточно сложными именно с точки зрения практической реализации. Невозможность заранее определить способ передачи данных и соответственно предусмотреть проблемы на этапе интеграции, а также оценить объем конечных данных и данных, пригодных для анализа, требует добавления дополнительных уровней очистки и подготовки данных. Таким образом, включение в алгоритм визуализации дополнительного шага позволяет сформировать буфер (витрину данных), который нивелирует возникающие проблемы и позволяет аналитику и особенно конечным пользователям иметь в распоряжении более широкий, чем формируется непосредственно с источника, и спектр данных для визуализации.

Данный алгоритм может быть использован не только для задач мониторинга ДТП. Он применим для визуализаций, где требуется интеграция данных с разными моделями, структурами и типами.

Литература

1. Федеральный закон от 10.12.1995 № 169-ФЗ (ред. от 08.12.2020) «О безопасности дорожного движения». – ст. 2.

2. *Doan, A.* Principles of Data Integration / A. Doan, A. Halevy, Z. Ives – 1st Edition – Burlington: Morgan Kaufmann, 2012. – 520 p.

3. *Reeve, A.* Managing Data in Motion: Data Integration Best Practices Techniques and Technologies / A. Reeve. – 1st Edition – Burlington: Morgan Kaufmann, 2013. – 204 p.

4. *Ухлова, В. В.* Основы технологий Big Data / В. В. Ухлова // Воронеж: Издательский дом ВГУ, 2020. – 81 с.

5. *Sherman, R.* Business Intelligence Guidebook: From Data Integration to Analytics / R. Sherman. – 1st Edition – Burlington: Morgan Kaufmann, 2014. – 510 p.

Максименко Юлия Александровна – студентка 4-го курса кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: mailyulia@list.ru

Ухлова Вера Владимировна (научный руководитель) – канд. физ.-мат. наук, доцент кафедры математических методов исследования операций Воронежского государственного университета. E-mail: it_content@list.ru

УДК 004.94

АНАЛИЗ МЕТОДОВ УДАЛЕНИЯ НЕВИДИМЫХ ПОВЕРХНОСТЕЙ ПРИ ВИЗУАЛИЗАЦИИ ТРЕХМЕРНЫХ СЦЕН

А. Ю. Масюков, Е. В. Трофименко

Воронежский государственный университет

Введение

Удаление невидимых поверхностей – один из основных методов ускорения визуализации сложных трёхмерных сцен в реальном времени [2]. Конечная цель методов удаления невидимых поверхностей – предотвратить отправку невидимых объектов в конвейер визуализации. Стандартный метод удаления невидимых поверхностей – это отбраковка по усечённой пирамиде вида камеры. Это быстрый и простой метод, но он не устраняет объекты в области видимости, которые перекрываются другими объектами. Это может привести к значительному расходу вычислительных мощностей, так как одна и та же область изображения будет обработана более одного раза. Проблема решается методами удаления невидимых поверхностей [7].

Существует два принципиально разных подхода обработки сцены: offline (предварительная) обработка сцены и online обработка (в реальном времени) [3]. Недостатки первого подхода заключаются в том, что требуется заранее просчитать потенциально видимые и невидимые объекты. Такой алгоритм довольно сложен в реализации, а также потребуются очень большие вычислительные мощности. Второй подход решает эти проблемы, добавляя дополнительные вычисления на каждый кадр в реальном времени. В статье будет рассматриваться только второй подход.

Современное графическое оборудование изначально поддерживает запросы на определение видимости объекта (occlusion query). Хотя сам запрос обрабатывается быстро с использованием исключительно мощности графического процессора (GPU), его результат не доступен сразу из-за задержки между выдачей запроса и его фактической обработкой в графическом конвейере. В результате простое применение запросов окклюзии может даже снизить общую производительность приложения из-за связанных с этим простоев центрального процессора и ожидания работы графического процессора.

1. Существующие методы решения задачи

В статье будут рассмотрены следующие методы:
- View Frustum Culling
- Hierarchical Stop-and-Wait Method
- Coherent Hierarchical Culling
- Near Optimal Hierarchical Culling
- CHC++: Coherent Hierarchical Culling Revisited

Для сравнения методов была создана трёхмерная сцена на языке C++ с использованием технологии OpenGL [1].

1.1. Метод 1: View Frustum Culling

Удаление невидимых поверхностей по усечённой пирамиде вида камеры (View Frustum Culling) – это процесс удаления из процесса рендеринга объектов, которые полностью лежат

за пределами видимости пирамиды вида [8]. Визуализация этих объектов будет пустой тратой времени, поскольку они не видны напрямую. Чтобы ускорить отбраковку, обычно используют ограничивающие объемы, окружающие объекты, а не сами объекты (рис. 1).

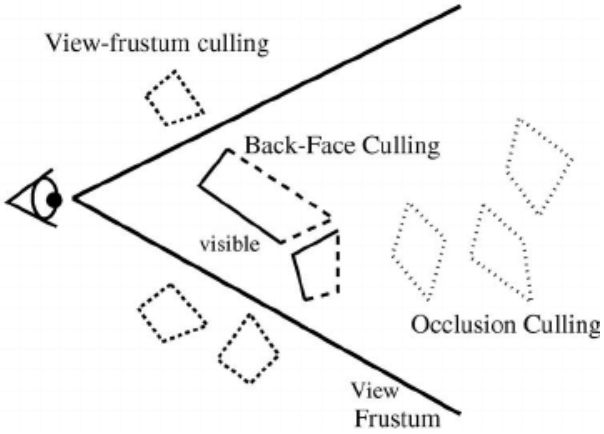


Рис. 1. Пример удаления невидимых объектов

1.2. Метод 2: Hierarchical Stop-and-Wait Method

Иерархический метод остановки и ожидания (Hierarchical Stop-and-Wait Method) работает следующим образом. Строится KD-дерево для трёхмерного пространства. После того, как узел KD-дерева проходит удаление по усечённой пирамиде вида камеры, он проверяется на перекрытие путем создания запроса на определение видимости объекта и ожидания его результата. Если узел оказывается видимым, мы продолжаем рекурсивно тестировать его дочерние элементы в порядке от начала до конца. Если узел является листом, мы визуализируем связанные с ним объекты [3].

Проблема с этим подходом заключается в том, что мы можем продолжить обход дерева только тогда, когда станет доступен результат последнего запроса видимости объекта. Если результат недоступен, процессор вынужден простаивать, что приводит к значительному снижению производительности (рис. 2). Эти простои вместе с накладными расходами самих запросов могут даже снизить общую производительность приложения по сравнению с простой отбраковкой по пирамиде вида.

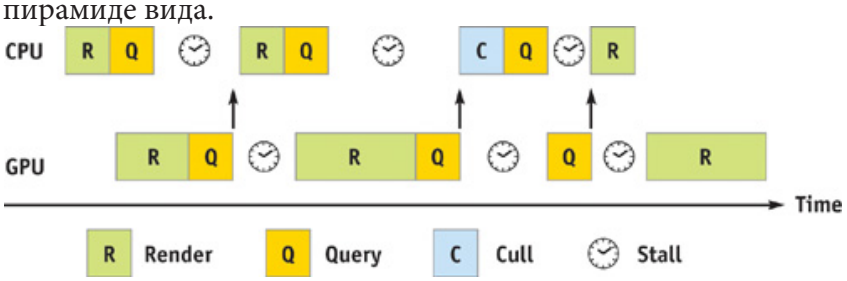


Рис. 2. Простои центрального и графического процессоров

1.3. Метод 3: Coherent Hierarchical Culling

Алгоритм когерентного иерархического отбора (Coherent Hierarchical Culling) использует временную и пространственную когерентность для уменьшения накладных расходов и задержки запросов аппаратной окклюзии. Он проходит по иерархии KD-дерева в порядке от начала до конца и выдает запросы только для ранее видимых листьев и узлов ранее невидимой границы [4]. Предполагается, что ранее видимые листья остаются видимыми в текущем кадре и, следовательно, визуализируются немедленно. Результат запроса для этих узлов обновляет

их состояние только для следующего кадра. Предполагается, что невидимые узлы остаются невидимыми, но алгоритм извлекает результат запроса в текущем кадре, чтобы обнаружить изменения видимости (рис. 3).

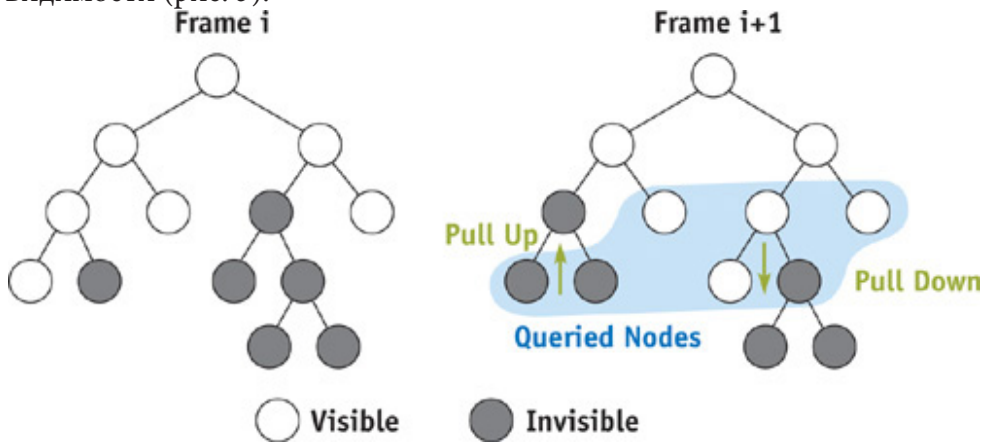


Рис. 3. Видимость узлов иерархии в двух последовательных кадрах

Уменьшение количества запросов (запросы не создаются для ранее видимых внутренних узлов) и их чередование снизили накладные расходы на запросы определения видимости объекта до приемлемого количества. Алгоритм очень хорошо работает в сценариях, в которых много преград. Однако на более новом оборудовании, где рендеринг геометрии становится дешевым по сравнению с запросом, или на точках обзора, где видна большая часть сцены, этот метод может стать даже медленнее, чем обычное удаление невидимых поверхностей по усечённой пирамиде вида камеры (View Frustum Culling). Это результат бесполезных запросов и ненужных изменений состояния. Эта проблема делает алгоритм CHC менее привлекательным для разработчиков игр, которые требуют алгоритма, который надежно быстрее, чем отбраковка по пирамиде вида. Другая проблема CHC заключается в сложной интеграции метода в цикл рендеринга высоко оптимизированных игровых движков. CHC чередует рендеринг и запросы отдельных узлов пространственной иерархии, что не позволяет механизму выполнять сортировку материалов и приводит к большему количеству вызовов API механизма [6].

На рис. 4 приведён график сравнения времени для обработки кадров для трёх методов: View Frustum Culling, Hierarchical Stop-and-Wait Method и Coherent Hierarchical Culling. Также есть так называемый «идеальный» алгоритм – который визуализирует видимые поверхности, при этом не выполняя проверки на видимость.

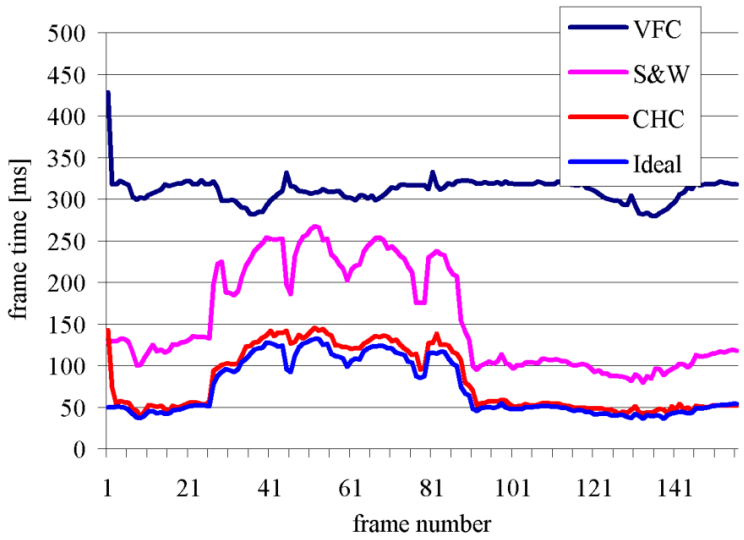


Рис. 4. Сравнение времени кадров разных методов для сцены

1.4. Метод 4: Near Optimal Hierarchical Culling

Алгоритм, близкий к оптимальному иерархическому выбору (Near Optimal Hierarchical Culling) решает проблему бесполезных запросов. Метод использует откалиброванную модель графического оборудования для оценки стоимости запросов и затрат на рендеринг. Он оценивает перекрытие узлов, используя простую модель покрытия экрана и дальнейшие поправки, предполагающие временную согласованность. Оценка окклюзии и аппаратная модель используются в эвристике затрат и выгод, которая решает, применять ли запрос на определение видимости к текущему обрабатываемому узлу [5].

Алгоритм сохраняет значительное количество запросов, особенно запросов, которые будут применяться к видимым узлам. Это может привести к значительному улучшению алгоритма CHC, если предполагаемая оптимизация видимости, предложенная для CHC, не используется.

Результаты для NOHC показывают, что при правильной аппаратной калибровке метод всегда работает лучше, чем отбраковка пирамиды обзора.

На рис. 5 приведён график сравнения времени для обработки кадров для двух методов: Coherent Hierarchical Culling и Near Optimal Hierarchical Culling. Также есть так называемый «оптимальный» алгоритм отбраковки на основе запросов определения видимости объекта. Оптимальный алгоритм выводится в предположении, что состояние каждого отбракованного узла должно быть проверено [4].

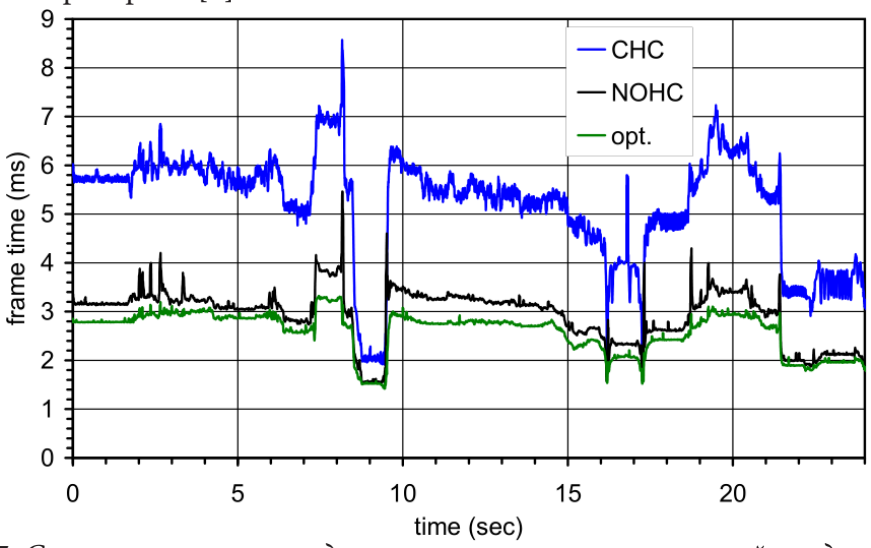


Рис. 5. Сравнение времени кадров при максимальных настройках драйвера

1.5. Метод 5: CHC++: Coherent Hierarchical Culling Revisited

Алгоритм CHC++ нацелен на сохранение простоты алгоритма CHC с несколькими важными дополнениями. Как и в CHC, используется приоритетная очередь для обхода иерархии. Эта очередь обеспечивает порядок обработки обрабатываемых узлов от начала до конца. В отличие от CHC, новый алгоритм использует две новые очереди для хранения узлов, которые следует запросить (v-очередь и i-очередь, рис. 6). Эти две очереди являются ключом к сокращению изменений состояния рендеринга и компиляции множественных запросов.

Ранее видимые узлы визуализируются немедленно, как в CHC. Если они запланированы для проверки в текущем кадре, они помещаются в v-очередь. Алгоритм планирования запросов использует описанный шаблон выборки с временным сдвигом для уменьшения количества запросов и их равномерного распределения по кадрам. Запросы узлов, хранящиеся в v-очереди, используются для заполнения времени ожидания, если оно должно произойти. В конце кадра оставшиеся узлы в v-очереди образуют единый пакет запросов.

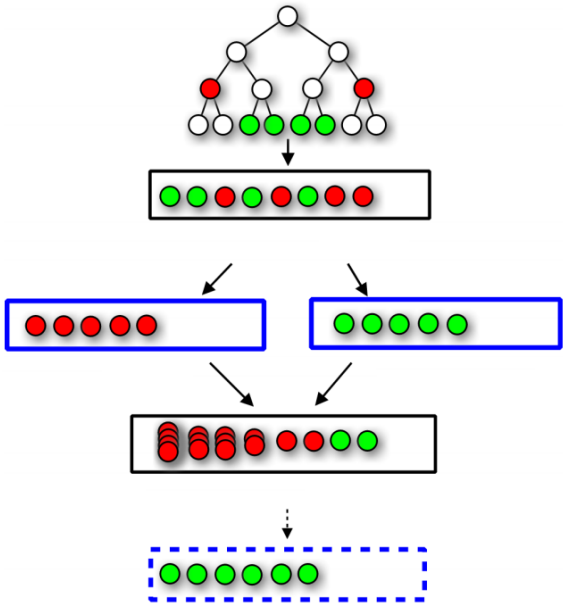


Рис. 6. Различные очереди, используемые алгоритмом CHC ++

В i-очередь накапливаются обработанные узлы, которые были невидимы в предыдущих кадрах. Когда в очереди имеется достаточное количество узлов, мы применяем пакет запросов на определение видимости объектов для узлов в i-очереди при их компиляции.

При интеграции метода в игровой движок видимые узлы сначала накапливаются в очереди рендеринга. Очередь рендеринга затем обрабатывается движком до того, как будет отправлен пакет запросов из i-очереди [4].

На рис. 7 приведён график сравнения времени для обработки кадров для четырёх методов: View Frustum Culling, Coherent Hierarchical Culling, Near Optimal Hierarchical Culling и CHC++. По графику хорошо видно, что алгоритм CHC++ намного быстрее обрабатывает кадры трёхмерной сцены.

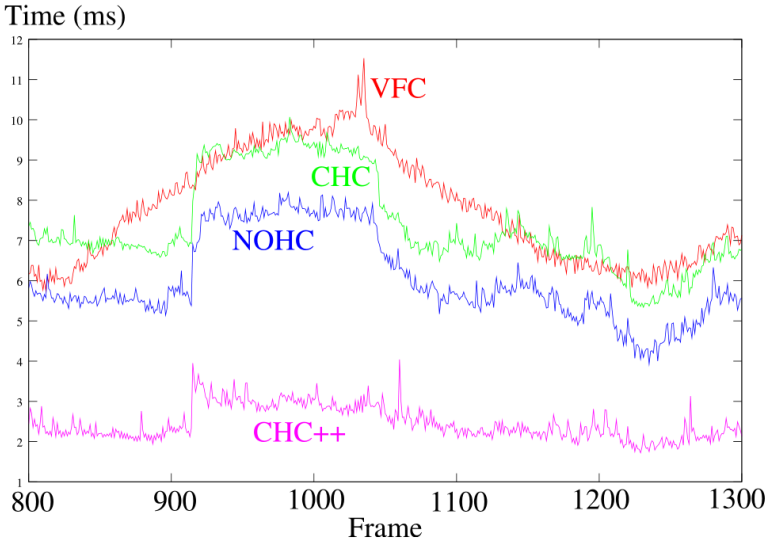


Рис. 7. Сравнение времени кадров четырёх методов

Заключение

Было продемонстрировано преимущество алгоритма CHC++, который практически исключает время ожидания результатов запросов на определение видимости объектов как на

CPU, так и на GPU. Это достигается за счет использования временной когерентности, предполагая, что объекты, которые были видны в предыдущем кадре, остаются видимыми в текущем кадре. Алгоритм также уменьшает количество дорогостоящих запросов на окклюзию за счет использования иерархии для отсеивания больших закрытых областей с помощью одной проверки. В то же время алгоритм избегает запросы на определение видимости объектов для большинства других внутренних узлов.

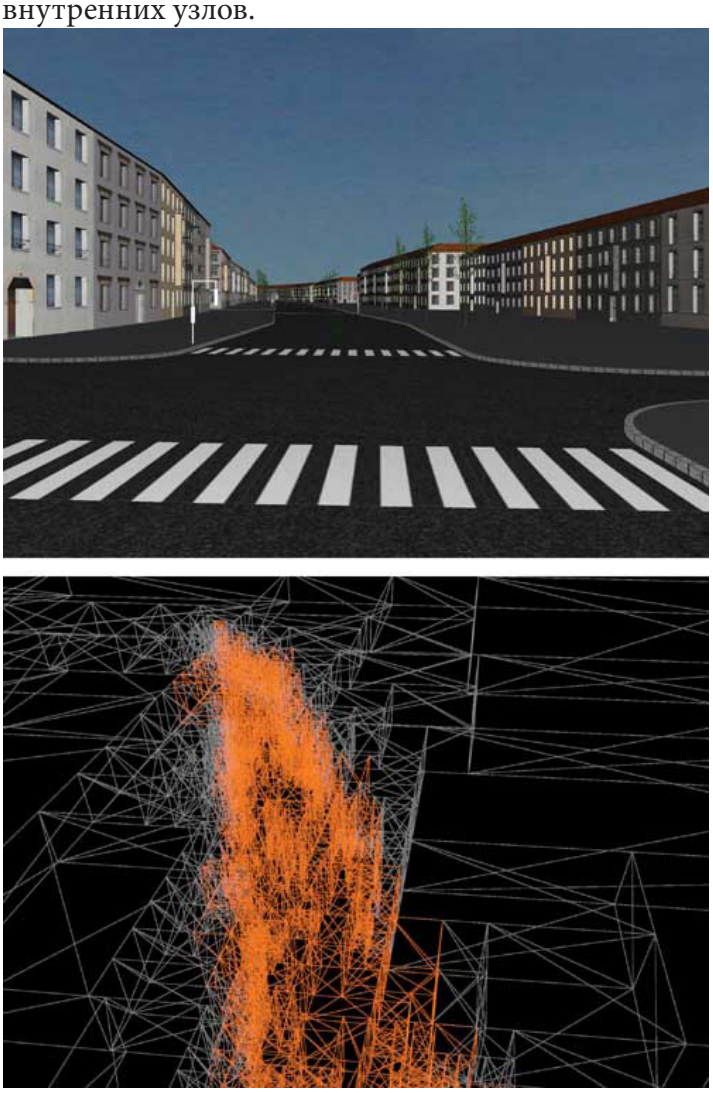


Рис. 8. Визуализация обработки сцены с помощью алгоритма CHC++

Этот алгоритм должен сделать аппаратные запросы на определение видимости объекта полезными для любого приложения, которое имеет большое количество окклюзий и где важны точные изображения. Например, на рис. 8 показано применение алгоритма к просмотру модели города с приведенной ниже классификацией видимости узлов иерархии. Оранжевые узлы были обнаружены видимыми; все остальные изображенные узлы невидимы. Обратите внимание на увеличивающийся размер невидимых узлов по мере увеличения расстояния от видимого множества.

Литература

1. *Gordon, V. S.* Computer Graphics Programming in OpenGL with C++ / V. S. Gordon, J. L. Clevenger. – Mercury Learning & Information, 2018. – 384 p.

2. *Bittner, J.* Visibility in computer graphics / J. Bittner, P. Wonka // Environment and Planning B: Planning and Design. – 2003. – Vol. 30, No 5. – P. 729–755.

3. Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful / J. Bittner [et al.] // Computer Graphics Forum. – 2004. – Vol. 23, No 3. – P. 615–624.

4. *Mattausch, O.* CHC++: Coherent Hierarchical Culling Revisited / O. Mattausch, J. Bittner, M. Wimmer // Computer Graphics Forum (Crete, 14–18 April 2008). – Crete, 2008. – Vol. 27, No 2. – P. 221–230.

5. *Guthe, M.* Near Optimal Hierarchical Culling: Performance Driven Use of Hardware Occlusion Queries / M. Guthe, Á. Balázs, R. Klein // EGSR '06: Proceedings of the 17th Eurographics conference on Rendering Techniques. (June, 2006). – P. 207–214.

6. *Wimmer, M.* Hardware Occlusion Queries Made Useful / M. Wimmer // Pharr M. GPU Gems 2: Programming Techniques For High-Performance Graphics And General-Purpose Computation / M. Wimmer, J. Bittner. – 2005. – P. 91–108.

7. Hidden-surface determination. – Режим доступа: https://en.wikipedia.org/wiki/Hidden-surface_determination

8. Viewing frustum. – Режим доступа: https://en.wikipedia.org/wiki/Viewing_frustum

Масюков Александр Юрьевич – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ факультета ПММ Воронежского государственного университета.
E-mail: masyukov@amm.vsu.ru

Трофименко Елена Владимировна (научный руководитель) – канд. физ.-мат. наук, доцент кафедры математического обеспечения ЭВМ факультета ПММ Воронежского государственного университета. E-mail: evtrof@gmail.com

УДК 004.94

СОЗДАНИЕ ИНТЕРАКТИВНОГО МЕНЮ РЕСТОРАНА С ЭЛЕМЕНТАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

А. Е. Мащенко

Воронежский государственный университет

Введение

Дополненная реальность (AR – Augmented Reality) позволяет накладывать виртуальные цифровые элементы поверх изображения реального пространства, взятого с камеры смартфона или планшета [1-2]. При этом пользователь может взаимодействовать с виртуальными объектами и получать ответную реакцию на свои действия.

AR отлично подходит для рекламных целей, позволяя визуализировать то, что покупатель не может увидеть вживую, для образования, обеспечивая получение знаний без использования дополнительного дорогостоящего оборудования, для навигации как внутри зданий, так и на открытой местности, а также – для развлечений и геймификации.

В последнее время интерес к дополненной реальности неуклонно растет. В связи с пандемией многие компании сталкиваются с необходимостью переноса бизнеса в онлайн. Использование современных информационных технологий может помочь им пережить в кризис, предлагая новые способы привлечения клиентов.

До недавнего времени дополненная реальность требовала создания отдельного мобильного приложения, в силу чего аудитория, использующая такое приложение, была ограничена – не все пользователи готовы тратить время на скачивание и установку приложения. Широкие возможности открыла дополненная реальность, реализованная в вебе. В этом случае доступ к AR-контенту могут получить практически все пользователи, при этом небольшим компаниям нет необходимости инвестировать в разработку отдельного приложения. Кроме того, веб-приложения являются кроссплатформенными, широко настраиваемыми и не требовательными к ресурсам и аппаратной платформе.

1. Реализация приложения дополненной реальности в вебе

Для демонстрации реализации дополненной реальности в браузере рассмотрено приложение для ресторанов и кафе, позволяющее дополнить обычное меню виртуальными изображениями отдельных блюд. Трехмерные модели блюд можно масштабировать и вращать, чтобы пользователи точно знали, как будет выглядеть заказанное блюдо. Кроме того, помимо внешнего вида блюда, пользователи могут посмотреть детальную информацию о его ингредиентах, весе, калорийности.

Полученный программный продукт базируется на использовании кроссплатформенного фреймворка Ar.js [3], дающего возможности отслеживать определенные маркеры в пространстве и накладывать изображения поверх них с помощью простого HTML-кода.

Для программы на основе Ar.js необходимо создать HTML-файл и импортировать одну из библиотек рендеринга: A-Frame или Three.js.

JavaScript библиотека Three.js специально разработана для 3D-рендеринга. Она предоставляет широкий набор инструментов для создания 3D-объектов, контроля над текстурами и сетками. Фреймворк A-Frame основан на Three.js, но не такой мощный и гибкий. Он подходит для использования в небольших проектах, позволяя добавлять объекты в сцену через HTML-

код, что улучшает наглядность иерархии сцены и делает разработку приложений быстрой и простой. Для простого демонстрационного приложения лучшим образом подходит A-Frame.

Далее необходимо создать сцену и указать в качестве источника изображения камеру мобильного устройства. После чего можно установить отслеживаемый маркер, при обнаружении которого будет появляться объект, и добавить 3D-модель.

Для добавления основных элементов используются теги <a-scene>, <a-marker>, <a-entity>. Полный HTML-код представлен в Листинге 1.

Листинг 1

```
<html>
<head>
<script src=»aframe.min.js»></script>
<script src=»aframe-ar.js»></script>
</head>
<body style='margin : 0px; overflow: hidden;'>
<a-scene embedded arjs='sourceType: webcam;'>
<a-marker type='pattern' url='assets/pattern-qr-code.patt'>
<a-entity
                        scale=»5 5 5»
                        obj-model=»obj: url(hamburger/Ham.obj);
                </a-entity>
                <a-entity
                        position=»0 1 0»
                        scale=»1 1 1»
                        obj-model=»obj: url(hamburger/Ham.obj);
                </a-entity>
                </a-marker>
<a-camera-static/>
</a-scene>
</body>
</html>
```

Маркер представляет собой квадратное изображение с черной рамкой и максимальным разрешением 16x16 пикселей. Для создания маркера можно воспользоваться утилитой AR.js Marker Training, которая позволяет получить файл паттерна маркера .patt и сам маркер в виде jpeg-изображения. За основу маркера взят QR-код. В дальнейшем он может быть использован для перехода на веб-страницу приложения.

Для использования Web-AR технологии необходимо запустить стандартный браузер и перейти по ссылке, после чего откроется веб-страница с возможностью воспроизведения контента дополненной реальности. Этот контент появится, как только в поле зрения камеры попадет предопределенный маркер. Воспроизводимым контентом в данном примере является 3D-модель блюда, созданная с помощью программы Blender [4], хотя в общем случае, это может быть любое изображение, звук, видео, анимация и т. д.

Работа полученного приложения продемонстрирована на рис. 1.

Заключение

Дополненная реальность в вебе – перспективная технология, только набирающая популярность. Ее основное преимущество – простота внедрения. Такие приложения могут быть использованы на любом мобильном устройстве, имеющем доступ к интернету, без установки дополнительного программного обеспечения. Это позволит сэкономить на разработке мобильных приложений и в то же время создать уникальный клиентский опыт.

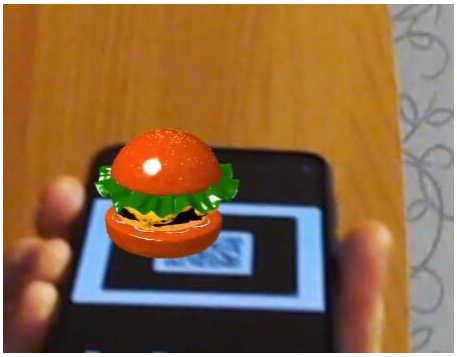
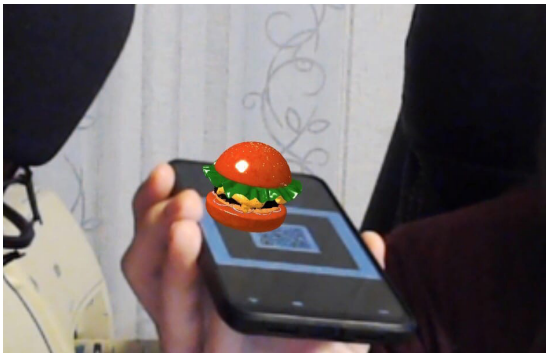



Рис 1. Демонстрация работы приложения

Литература

1. *Guazzaroni, G.* Virtual and Augmented Reality in Education, Art, and Museums (Paperback) / G. Guazzaroni, A. S.Pillai. - IGI Global: United States, 2019.

2. *Barfield, W.* Research Handbook on the Law of Virtual and Augmented Reality / W. Barfield [et. al]. – Edward Elgar Pub, 2020. - 712 p.

3. AR.js – Augmented Reality on the Web : [сайт]. – URL: https://ar-js-org.github.io/AR.js-Docs/ (дата обращения: 12.03.2021).

4. Blender 2.90 Reference Manual : [сайт]. – URL: https://docs.blender.org/manual/en/latest/index.html (дата обращения: 10.04.2021).

Мащенко Александр Евгеньевич – студент 2-го курса Воронежского государственного университета. E-mail: mashchenko-st@mail.ru

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: Bolotova.svetlana@gmail.com

УДК 004.6, 331.108

ТЕХНОЛОГИИ DATA MINING В ПРОГНОЗИРОВАНИИ ОТТОКА ПЕРСОНАЛА

В. В. Морозова, В. В. Ухлова

Воронежский государственный университет

Введение

На сегодняшний день персонал представляет собой один из основных ресурсов организаций. Соответственно, эффективное управление им влияет на состояние и развитие компаний. При этом, одной из проблем работы с персоналом является текучесть кадров. От нее напрямую зависит уровень квалифицированных сотрудников, косвенно – такой показатель как прибыль. Как следствие, текучесть персонала очень дорогостоящий процесс для организации. Чтобы снизить риски ухода сотрудников, HR-менеджерам приходится внедрять практики для сохранения рабочих команд. Ричард Розенов в своей статье [1] подчеркивает, что стандартные методы перестают эффективно справляться с данной функцией, поэтому все более распространяются технологии называемые термином «data mining». Возможность их применения к процессам, связанным с текучестью персонала, обоснована тенденцией увеличения количества собираемой информации о сотрудниках и расчета различных показателей функционирования организации. В статье рассмотрена задача выявления закономерностей в данных для управления оттоком сотрудников методами технологий «data mining», которая в настоящее время актуальна для многих организаций.

1. Постановка задачи

В зависимости от отрасли компании и рода деятельности конкретных специалистов уровень текучести может существенно варьироваться. Согласно статистике рекрутинговой компании Antal Russia (рис.1), самыми подверженными оттоку являются сфера розничной торговли, медиа отрасль и индустрии развлечений.

Рис. 1. Статистика оттока сотрудников рекрутинговой компании Antal Russia

Так в рознице показатель текучести составляет 48 %, а в медиа и индустрии развлечений 43 %. На третьем месте по оттоку сотрудников в рейтинге стоят компании по производству упаковки и бумаги – 38 %.

В работе ставится задача нахождения показателей, влияющих на склонность сотрудников к оттоку. На данных показателях в дальнейшем планируется построить модель прогнозирования оттока сотрудников. Последняя, по мнению авторов, позволит своевременно принимать меры по снижению рисков оттока сотрудников и подбора мотивационных мер для сохранения «ценных» для организации сотрудников.

2. Алгоритм исследования

В качестве инструментария выбраны методы технологии «data mining». Они позволяют производить интеллектуальную обработку данных, т.е. с использованием методов машинного обучения, математической статистики и теории баз данных. В качестве инструмента работы с данными был выбран язык программирования Python.

2.1. Сбор данных

В исследовании использовались данные о сотрудниках отдельно взятой организации за период, равный 6 месяцев. При этом сотрудники изначально были разделены на группы уже уволившихся и работающих на данный момент. Данные были обезличены и представлены набором таблиц.

Выборка исследования содержит следующие данные:

1) социально-демографические данные (пол, возраст, регион проживания, доход, образование);

2) транзакционные данные на рабочих устройствах;

3) принадлежность к различным группам внутренней сегментации компании.

2.2. Предобработка данных

На этапе предобработки данных, был выполнен анализ, который позволил выявить ошибки ввода, выбросы и пустые данные. В качестве предобработки данных использовались два основных метода:

1) фильтрация переменных с низким значением дисперсии;

2) удаление выбросов с использованием Isolation Forest.

Первый метод необходим для уменьшения размерности матриц, получаемых после применения первого метода. После преобразования категориальных переменных полученная матрица получается слишком большой и разряженной. Чтобы уменьшить размерность матрицы, переменные, у которых встречается в 95 % случаях одинаковое значение, удаляются. Для этого применяется простая статистическая фильтрация с пороговым значением дисперсии. Также, для увеличения точности модели, из обучающих данных были убраны выбросы.

Второй метод – фильтрации выбросов. В исследовании данных используется алгоритм нахождения выбросов с помощью ансамбля бинарных деревьев Isolation Forest. Выбросом может считаться запись о сотруднике, в которых имеют место значения некоторых предикторов, сильно больше или сильно меньше, чем средние значения в выборке.

После предобработки данных были получены простейшие закономерности.

2.3. Анализ и моделирование данных

Напомним, что изначально полученная база сотрудников была разделена на группы уволившихся и текущих. Текущий сотрудник определялся как работник, который продолжил работать на момент скоринга еще минимум три месяца. Также были собраны статистические

данные в разрезе имеющихся показателей, что позволило определить набор характеристик, которые могут оказывать влияние на увольнение сотрудников. Результаты по наиболее выделившимся показателям представлены в табл. 1. Для вышеупомянутых групп сотрудников было рассчитано процентное распределение значений по показателям.

Таблица 1

Статистические результаты экспериментов

Название показателя	Группа уволившихся	Группа работающих
JOB_INTEREST_IND	1 – 46 % 0 – 54 %	1 – 18 % 0 – 82 %
INACTIVE_CAT	Низкий – 8 % Средний – 65 % Высокий – 27 %	Низкий – 2 % Средний – 56 % Высокий – 42 %
ACTIVE_CREDIT_IND	1 – 13 % 0 – 87 %	1 – 23 % 0 – 77 %
AGE_CAT	< 25 – 23 % 25-45 – 76 % 45-55 – 1 % > 55 – 0 %	< 25 – 9 % 25-45 – 87 % 45-55 – 3 % > 55 – 1 %

Для предсказания вероятности оттока текущих сотрудников был выбран метод логистической регрессии. Логистическая регрессионная модель предназначена для решения задач предсказания дискретной зависимой переменной. В силу того, что в модели присутствует гладкая монотонная возрастающая нелинейная функция (функция сигмоиды), появляется возможность получить непрерывные значения зависимой переменной (от 0 до 1). Это становится возможным в случае бинарной классификации. В анализе данных ее часто используют для предсказания вероятности наступления некоторого события в зависимости от значений имеющихся предикторов. В рамках текущей задачи за счет этого можно получить вероятность склонности сотрудника к оттоку.

3. Результаты исследования

Результаты работы логистической регрессионной модели представлены в рис. 2.

Logit Regression Results

Dep. Variable:	IsChurned	No. Observations:	40000
Model:	Logit	Df Residuals:	39995
Method:	MLE	Df Model:	5
Date:	Sat, 13 Feb 2021	Pseudo R-squ.:	0.6091
Time:	10:21:55	Log-Likelihood:	-11935.
converged:	True	LL-Null:	-13863.
	LLR	p-value:	0.0074

	coef	std err	z	P>|z|
const	-2.1417	0.050	-43.216	0.000
job_interest_ind	1.2545	0.001	32.013	0.
inactive_cat	0.245	0.123	2.465	0.0953
active_credit_ind	-1.961	0.104	20.4	0.001
age_cat	1.1137	0.504	5.64	0.0854

Рис. 2. Вывод коэффициентов показателей

Анализ значений полученных коэффициентов позволяет сделать вывод, что на предсказание оттока влияют:

1) показатели интереса к поиску работы;

2) активные кредитные продукты.

Для тестирования были использованы ретро данные по сотрудникам с индикаторами факта увольнения. После получения результатов предсказаний модели была определена точность 78 %.

Заключение

Полученная в ходе исследования модель требует уточнения. Однако уже на данном этапе использования она позволяет сделать выводы о причинах оттока сотрудников и разрабатывать меры для сохранения значимых для организации сотрудников. Также в ходе исследования, изучение отчетности дало возможность установить общие характеристики между сотрудниками, заинтересованными в смене работы.

Таким образом, анализ на основе методов технологий «data mining» помог найти показатели, влияющие на склонность сотрудников к оттоку, на которые работодатели (служба персонала) обычно не обращают внимания, а разработанную модель можно использовать для прогнозирования рисков потери работников. Как результат, использование данной технологии помогает создавать более эффективную и целостную кадровую стратегию, направленную на удержание специалистов.

Литература

1. Analyzing Employee Turnover – Descriptive Methods. – Режим доступа: https://www.linkedin.com/pulse/analyzing-turnover-descriptive-methods-richard-rosenow-pmp (дата обращения: 18.04.2021).

3. Data mining технология. – Режим доступа: https://iot.ru/wiki/data-mining (дата обращения: 18.04.2021).

4. Основы технологий Big Data: учебное пособие / Составитель: В. В. Ухлова; Воронежский государственный университет. – Воронеж : Издательский дом ВГУ, 2020. – 81 с.

Морозова Валерия Владимировна – студентка 2-го курса факультета прикладной математики, информатики и механики Воронежского государственного университета.
E-mail: lera.morozova.1997@mail.ru

Ухлова Вера Владимировна (научный руководитель) – канд. физ.-мат. наук, доц. кафедры математических методов исследования операций Воронежского государственного университета. E-mail: it_content@list.ru

УДК 004.65

1С: «ДИСТАНЦИОННОЕ ОБУЧЕНИЕ»

П. С. Науменко

Воронежский государственный университет

Введение

Из-за сложившейся эпидемиологической обстановки в стране, многие организации и государственные учреждения перешли на дистанционное обучение. Дистанционное обучение – обучение, при котором все или большая часть учебных процедур осуществляется с использованием современных информационных и телекоммуникационных технологий при территориальной разобщенности преподавателя и студентов. Существует множество приложений и сервисов, предназначенных для удаленной работы. В данной статье речь пойдет о программном продукте 1С: «Дистанционное обучение», написанном на платформе 8.3.

1. 1С: «Дистанционное обучение»

1С: «Дистанционное обучение» предназначено для связи между учениками и преподавателями, проведения видео уроков и тестирований, а также анализа результатов и подведение итогов обучения.

В данной конфигурации есть разделение на роли «ученик» и «преподаватель», учеников можно разбить на группы и составить каждой группе расписание занятий.

В конфигурации реализована возможность создавать курсы, в которые входят: лекции, ссылки на материалы (презентации, книги, видеоуроки), задания, тесты. Ссылки на пособия, задания и прочие материалы можно прикрепить в любом формате.

Ученик на своем рабочем столе может видеть свою ближайшую задачу (например, «сдать задание № 2 по курсу “курс_3”»). Он может просматривать материалы тех курсов, к которым у него есть доступ. Подключаться к видеоконференциям по своему расписанию. Так же ему доступны некоторые отчеты, по которым он может следить за своей успеваемостью, просмотреть все свои задолженности и срок их сдачи.

У преподавателя есть возможность создавать курсы. Добавлять в них разделы, материалы, задания (со сроком выполнения или без ограничений по времени), тестирования, а также организовывать видеосвязь. Каждый ответ ученика, он может прокомментировать и, если это необходимо, открыть возможность отредактировать ответ. В конце курса преподаватель может на основании отчета успеваемости поставить оценку ученику. Также преподавателю доступны отчеты, которые позволяют отслеживать активность ученика, количество посещений и пропусков видеоконференций, успеваемость обучающегося, итоги тестов. Все отчеты могут быть выгружены из программы.

В конфигурации 1С: «Дистанционное обучение» реализовано общение между учеником и преподавателем с помощью WhatsApp. Преподаватель может сделать рассылку сообщений всем своим группам, выбрать только одну группу или отправить нескольким ученикам. Также подключена возможность общения с помощью почты.

Также разработано мобильное приложение для данной конфигурации, которое повторяет весь функционал 1С: «Дистанционное обучение». Есть возможность настроить синхронизацию мобильного приложения с базой 1С.

2. Сравнение программ

В табл. 1 приведено сравнение конфигурации 1С: «Дистанционное обучение» с популярной бесплатной платформой moodle и платным аналогом iSpring.

Таблица 1

Сравнение программ

	1С: «Дистанционное обучение»	Moodle	iSpring
Быстрота внедрения	+	–	+
Интуитивно понятный дизайн	+	–	–
Масштабируемость	+	+	+
Поддержка мобильного приложения	+	+	+
Интеграция с другими сервисами и приложениями	+	+	+
Объем хранилища	+	+	–
Стоимость внедрения	–	+	–
Безопасность данных	+	+	+
Техподдержка	+	–	+

Заключение

Конфигурация 1С: «Дистанционное обучение», по сравнению со своими аналогами, имеет более понятный интерфейс, что уменьшает время на обучения персонала на работу с программным продуктом. Данную конфигурацию можно внедрить достаточно быстро и имеет техническую поддержку.

Литература

1. *Байдаков, В.* Руководство разработчика: Документация / В. Байдаков. В. Дранищев. А. Краюшкин; Часть 1. – М. : Изд-во Фирма «1С», 2009. – 638 с.

2. *Байдаков, В.* Руководство разработчика: Документация / В. Байдаков. В. Дранищев. А. Краюшкин; Часть 2. – М. : Изд-во Фирма «1С», 2009. – 640 с.

3. *Радченко, М. Г.* Пособие разработчика: Документация / М. Г. Радченко, Е. Ю. Хрусталева. – М. : Изд-во «1С-Паблишинг», 2013. – 964 с.

Науменко Павел Сергеевич – магистрант 2-го года обучения кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: naumenko.pvl.s@mail.ru

Кенин Сергей Леонидович (научный руководитель) – доцент кафедры ERP-систем и бизнес процессов Воронежского государственного университета.

УДК 519.8

ОБЪЕКТИВНОЕ ОЦЕНИВАНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ СТУДЕНТОВ ВУЗОВ В УСЛОВИЯХ ПРИМЕНЕНИЯ ДИСТАНЦИОННЫХ ТЕХНОЛОГИЙ

С. А. Палкина

Воронежский государственный университет

Введение

Повышение качества образования в российских вузах является на сегодняшний день приоритетной задачей министерства образования. Вводимые новые федеральные государственные образовательные стандарты теперь делают акцент на формирование у обучающихся определенного набора компетенций в соответствии с направлением подготовки. При этом процесс обучения явно разделился на самостоятельную и аудиторную работу. Более того, сложная эпидемиологическая ситуация в стране привела к обязательному применению в вузах дистанционных технологий. Как следствие, процесс оценки результатов обучения потребовал пересмотра методов контроля качества образования в части критериев освоении дисциплин. В данной работе рассмотрен один из подходов учета специфики применения дистанционных технологий в оценивании результатов освоения учебных дисциплин.

1. Постановка задачи

В настоящее время в вузах России широко используется балльно-рейтинговая система. Она заключается в формировании накопительной оценки успешности освоения дисциплины на основании совокупности использования контрольно-измерительных материалов. Баллы выставляются по результатам проверки знаний (умений, навыков) с применением контрольно-измерительного материала. Далее формируется итоговая суммарная оценка. Итоговая оценка предполагает перевод в традиционную 4-х балльную шкалу («отлично», «хорошо», «удовлетворительно», «неудовлетворительно») или 2-х балльную («зачтено» «не зачтено»). В рамках новых стандартов эту оценку следует понимать как уровень освоения дисциплины в целом, т. е. рейтинг. Такая система позволяет производить оценивание результатов обучения в вузах по всем видам занятий, используемым в учебном процессе. Однако при таком подходе нет разделения на виды занятий, например, практические занятия и лабораторные работы. При этом они могут иметь разный уровень значимости в освоении дисциплины. Более того, вид занятий определяет спектр применяемых контрольно-измерительных материалов (КИМ).

При использовании большого спектра КИМ и для учета вида занятий решением проблемы оценивания является применение интегральной итоговой оценки (1):

$$Q_{итог} = K_{пр}Q_{пр} + K_{лр}Q_{лр} + K_{тест}Q_{тест}, \quad (1)$$

где $K_{пр}$, $K_{лр}$, $K_{тест}$ – коэффициенты (веса) значимости практических работ, лабораторных работ и теста соответственно; $Q_{пр}$, $Q_{лр}$, $Q_{тест}$ – оценка, полученная за практическую лабораторную работу и тест.

В этом случае, веса значимости позволяют более объективно, в сравнении с 4-х балльной системой оценить уровень освоения дисциплины. При этом количество элементов оценивания и вес каждого определяется преподавателем индивидуально. Такой метод также позволяет учесть и форму проводимых занятий, например, дистанционные или очные занятия. С одной стороны, рассмотренная процедура прозрачна и объективна, с другой, при наличии большого

количества видов занятий и широкого спектра контрольно-измерительных материалов трудоемка относительно того, что важно объективно подобрать веса значимости каждого КИМ. В связи с этим, задача формирования шкалы весовых коэффициентов для балльно-рейтинговой системы оценки является актуальной.

2. Процедура формирования шкалы весовых коэффициентов

Формирование шкалы весовых коэффициентов для балльно-рейтинговой системы заключается в расчете величины значимости каждого из КИМов. Соответственно, ставится задача разработать модель оценивания значимости КИМ с учетом форм и видов учебных занятий, используемых в учебном процессе. В подходе, изложенном в данной работе, для этого использован метод анализа иерархии (МАИ) Он позволяет при формировании модели оценки учесть несколько факторов одновременно, при этом использовать качественные оценки. Согласно подходу МАИ, построим иерархию, которая повторяет процесс принятия решения при оценке значимости КИМ. Все виды занятий вуза разделяются на лекции, практические занятия (семинары) и лабораторные работы. Каждый из них может проходить в дистанционной и традиционной (оффлайн) аудиторной форме. Соответственно, иерархия имеет вид, представленный на рис. 1.

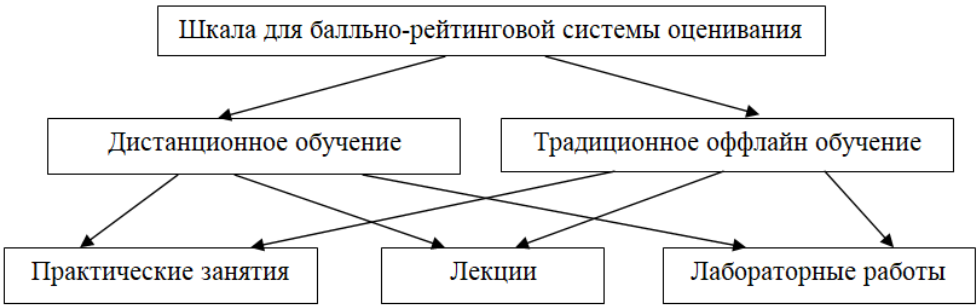


Рис. 1. Иерархия форм и видов занятий учебного процесса

Для каждого из видов занятий предусматривается свой набор оценочных средств, применяемых для оценки знаний, умений и навыков достижения компетенций осваиваемых дисциплин. Соответственно, следующим уровнем иерархии спектр оценочных средств. В свою очередь, для оценочных средств можно указать набор возможных КИМ (рис. 2).

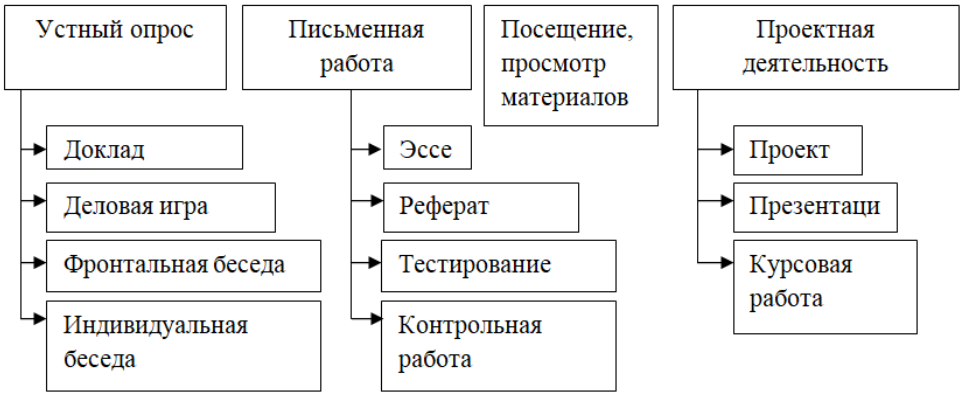


Рис. 2. Иерархия средств и материалов измерений

Синтез этих двух иерархий (рис. 1 и 2) позволяет получить итоговую модель оценки для формирования шкалы весовых коэффициентов для КИМ, используемых при расчете интегральной оценки (1). Итоговая модель учитывает форму занятий, их вид, средства измерений уровня достижимости компетенций дисциплины и вариант используемого КИМ (рис. 3).

Рис. 3. Модель оценивания значимости КИМ

На основе полученной модели формируется видение эксперта относительно значимости КИМ. Эксперт оценивает значимость элементов модели по уровням, начиная с самого верхнего. Оценка может трактоваться как приоритет и быть, например, 1, 2, 3 и т. д. или возможно назначение каждому элементу веса в общем наборе, например, в процентном соотношении, или выбирается вес с учетом шкалы Т.Саати. Особенностью применения метода МАИ является то, что эксперт при присвоении веса значимости сравнивает элементы иерархии попарно. Например, что значимее лекции или практические занятия, лекции или лабораторные работы.

Полученные оценки заносятся в матрицы парных сравнений, по которым вычисляются локальные веса элементов иерархии (значимости элементов на одном уровне). Далее проводится свертка весов по всей иерархии и рассчитываются глобальные веса каждого из уровней. Полученное множество значений глобальных весов самого нижнего уровня ранжируется и в итоговом варианте представляет собой шкалу коэффициентов (весов) значимости КИМ. Она и используется в качестве шкалы весовых коэффициентов для интегральной оценки балльно-рейтинговой системы.

Заключение

Данная модель оценки может применяться для интегральных оценок, где не все веса из иерархии используются. В этом случае им присваивается вес равный нулю. Кроме того, модель позволяет преподавателю объективно оценить значимость каждого из КИМ и в случае необходимости сокращения спектра их использования, подбирать комплект КИМ, позволяющий наиболее объективно оценить уровень освоения дисциплины.

Полученная модель учитывает все основные виды и формы занятий, используемые в образовательном процессе вузов России. При этом для учета специфики отдельных учебных заведений может быть расширена как по горизонтали, так и вертикали. При этом общий алгоритм формирования шкалы не измениться. В случае необходимости уменьшить количество элементов модели вследствие их неиспользования в вузе, модель не требует корректировки. При формировании видения эксперта «ненужные» элементы будут иметь самый малый вес.

Литература

1. *Азарнова, Т. В.* Метод анализа иерархий как средство поддержки принятия решений в стратегическом аналитическом планировании / Т. В. Азарнова, О. Ю. Пономарева, В. В. Ухлова // Экономическое прогнозирование: модели и методы: сб. тр. IX международной науч.-практ. конф. (Воронеж, 26 апреля 2013 г.). – Воронеж, 2013. – С. 9–12.

2. *Макаров, И. М.* Теория выбора и принятия решений / И. М. Макаров [и др.]. – Москва : Наука. 1982. – 330 с.

3. *Палкина, С. А.* Адаптация метода анализа иерархий к задаче оценки достижений учащихся / С.А. Палкина // Математические модели современных экономических процессов, методы анализа и синтеза экономических механизмов; Актуальные проблемы и перспективы менеджмента организаций в России: [сб. ст.] XIII Всерос. науч.-практ. конф. / Ин-т проблем упр. им. В. А. Трапезникова Рос. Акад. Наук, Самар. нац. исслед. ун-т им. С. П. Королева; гл. ред. Д. А. Новиков. – Самара : Изд-во СамНЦ РАН, 2020. – С. 81–87.

4. *Палкина, С. А.* Формирование шкалы коэффициентов значимости средств измерений для оценивания уровня освоения учебных дисциплин в высших учебных заведения / С. А. Палкина, В. В. Ухлова // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 7–9 декабря 2020 г. – Воронеж, 2021. – С. 1022–1026.

5. *Саати, Т.* Принятие решений : Метод анализа иерархий / Т. Саати; Пер. с англ. Р. Г. Вачнадзе. – Москва : Радио и связь, 1993. – 314 с.

Палкина София Александровна – студентка 2-го курса факультета прикладной математики, информатики и механики Воронежского государственного университета. E-mail: mail.soffya@mail.ru

Ухлова Вера Владимировна (научный руководитель) – канд. физ.-мат. наук, доцент кафедры математических методов исследования операций Воронежского государственного университета. E-mail: it_content@list.ru

УДК 004.8

ПРОГНОЗИРОВАНИЕ СПРОСА НА ОБЪЯВЛЕНИЕ НА САЙТЕ AVITO

Д. А. Петрова

Воронежский государственный университет

Введение

Сегодня виртуальные доски объявлений доступны каждому, у кого есть доступ к сети Интернет. Они могут иметь как узкую специализацию, принимая объявления только конкретных тематик (одежда, мебель и так далее), так и широкую направленность, охватывая большое количество разделов и освещая практически все сферы деятельности. В настоящий момент в сети существует свыше десяти тысяч различных сайтов для объявлений и все они отличаются по размеру, функционалу и способу взаимодействия с пользователями. Однако их суть сводится к одному: чтобы разместить объявление, каждый человек может зайти на сайт и заполнить соответствующую форму, указав свои контакты, текст объявления, при необходимости, уточнив некоторые условия или пожелания, а также приложить изображения для лучшего понимания сути.

При продаже товаров в Интернете описание товара может иметь большое влияние на интерес покупателя. В то же время, даже при подробном описании продукта и активном его продвижении, спрос на него может просто не существовать, что расстраивает продавцов, которые, возможно, слишком много инвестировали в маркетинг. Отсюда возникает вопрос: какие факторы влияют на статистику продаж товаров по объявлениям в интернете и каким образом?

Многие компании активно применяют технологии машинного обучения для повышения продаж: интернет-магазин Wildberries разрабатывает рекомендации для почтовых рассылок, а сеть «Перекресток» анализирует данные о покупателях, частоте и сумме их покупок, стиле жизни, приемлемом уровне цен, любимых категориях товаров.

Цель предлагаемого исследования заключается в построении эффективной модели для прогнозирования вероятности продажи товара на виртуальной доске объявлений и определении наиболее значимых признаков, влияющих на эту вероятность. В рамках исследования проведен первичный анализ и обработка исходных данных, сформированы новые признаки, построено несколько моделей машинного обучения для прогнозирования целевой переменной, найдена наиболее эффективная модель и проинтерпретированы ее результаты за счет выделения наиболее значимых факторов, влияющих на продажи, и определения характера этого влияния. Описанные задачи реализованы на языке программирования Python с помощью различных его библиотек на платформе для интерактивных вычислений Jupyter Notebook.

1. Исходные данные

1.1. Анализ данных

Для исследования используются данные, предоставленные в рамках конкурса крупнейшим интернет-сервисом для размещения объявлений в России – Avito. В своем соревновании Kaggle [1] Avito предлагает спрогнозировать спрос на товар объявления на основе его полного описания (заголовок, описание, изображения и так далее), географического контекста и исторического спроса для похожих объявлений в аналогичном контексте. Обладая этой информацией, Avito может проинформировать продавцов о том, как лучше всего оптимизировать их

описание товаров, и дать некоторое представление о том, какова будет вероятность продажи по данному объявлению.

Всего исходная выборка содержит 1503424 уникальных объявлений, каждое из которых описывается признаками, приведенными в табл. 1.

Таблица 1

Признаки объявлений

Тип признаков	Наименования признаков
Вещественные	Цена, количество объявлений продавца, дата объявления, рейтинг изображения
Категориальные	Номер продукта, номер пользователя, регион, город, категория продукта, подкатегория продукта, параметр 1, параметр 2, параметр 3, тип продавца
Текстовые	Заголовок объявления, описание объявления

Рейтинг изображения – это показатель качества и информативности изображения в объявлении, который определяет компания Avito. Признаки «Параметр 1», «Параметр 2» и «Параметр 3» – опциональные критерии для дополнительного, уточняющего описания продукта, являются особенностью сайта Avito. Для каждой подкатегории продукта они разные, также могут встречаться пустые значения (в случае, если пользователь не стал указывать параметр или товар вовсе не описывается какими-либо измерениями). Обобщенная модель описания товара Avito имеет некоторую иерархию: категория товара (личные вещи, транспорт, недвижимость, животные), подкатегория товара (одежда и обувь, детская одежда и обувь, часы и украшения), параметр 1 (мужская одежда, женская одежда, аксессуары), параметр 2 (брюки, верхняя одежда, джинсы и так далее), параметр 3 (размер, состояние, бренд). На рис. 1 приведен пример реализации параметров на сайте.

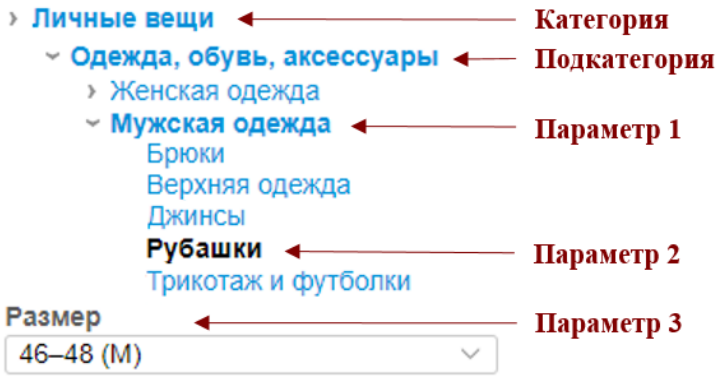


Рис. 1. Поиск товара на сайте Avito с помощью параметров

Целевая переменная – вероятность сделки, описывает шанс продажи по заданному объявлению. Таким образом, значение прогноза будет варьироваться в промежутке от нуля до единицы. Описательная статистика целевой переменной приведена в табл. 2.

Визуальный анализ данных проводился с помощью библиотек Matplotlib и Seaborn [2]. Для наглядности была построена гистограмма распределения целевой переменной (рис. 2). С помощью нее можно отметить, что распределение целевой переменной преобладает в значениях, близких к нулю, имеет заметную асимметрию с правосторонней скошенностью и некоторый «всплеск» значений около 0.8.

Если построить точечный график (рис. 3), отражающий зависимость целевого признака от цены, то можно прийти к выводу, что чем дороже объект объявления, тем меньше шансов, что он будет продан.

Таблица 2

Описательная статистика целевой переменной

Характеристика	Значение
Количество	1503424
Среднее значение	0.139
Стандартное отклонение	0.260
Минимум	0
Нижний квартиль	0
Медиана	0
Верхний квартиль	0.151
Максимум	1

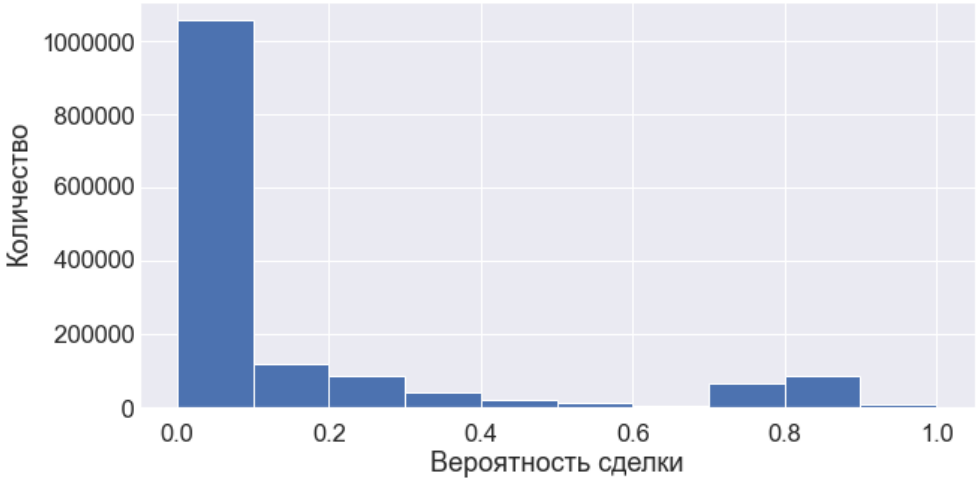


Рис. 2. Гистограмма распределения целевой переменной

На основе исходных данных были сформированы такие дополнительные признаки, как средний рейтинг продаж по региону, городу, категориям продукта, подкатегориям продукта и по каждому из трех параметров. Средний рейтинг продаж для категориального признака – это среднее арифметическое значений целевой переменной (вероятности продать товар) по данному категориальному признаку.

Было выявлено, что рейтинг продаж сильно зависит от категории, подкатегории продукта и его параметров. Наиболее важным признаком оказалась категория продаваемого продукта. На рис. 4 видно, что успешнее всего продаются услуги, транспорт и животные.

Рис. 3. Зависимость целевого признака от независимой переменной «Цена»

Рис. 4. Средняя вероятность сделки по каждой категории продукта

1.2. Обработка данных

Перед реализацией моделей машинного обучения была произведена проверка данных на пропуски и их заполнение. Текстовые признаки необходимо было очистить от лишних символов и стоп-слов (шумовых слов – слов, не несущих смысловой нагрузки в тексте), затем привести к виду цифрового вектора с помощью метода TfidfVectorizer. Метод TF-IDF (term frequency – inverse document frequency или частотность терминов – обратная частотность документов) позволяет избежать несоответствий, когда количество употреблений каждого слова в документе зависит от общего количества слов в документе [3].

Для работы моделей с категориальными признаками их тоже необходимо преобразовать в числовой тип данных. Для этого применялся метод One-Hot encoder. Исключением является модель машинного обучения CatBoost, она умеет работать с категориальными признаками без кодировки, поэтому для нее мы используем категориальные признаки без изменения.

2. Построение моделей

Для решения задачи было выбрано несколько моделей, основанных на градиентном бустинге деревьев решений. Градиентный бустинг – это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений [4].

В общем виде его шаги можно описать как:

1) построение простых моделей и анализ ошибок;

2) определение точек, которые не вписываются в простую модель;

3) добавление моделей, которые обрабатывают сложные случаи, которые были выявлены на начальной модели;

4) сбор всех построенных моделей, определяя вес каждого предсказателя.

Ансамбль – это набор предсказателей, которые вместе дают ответ (например, среднее по всем результатам). Причиной использования ансамблей является тот факт, что несколько предсказателей, которые пытаюсь получить одну и ту же переменную дадут более точный результат, нежели одиночный предсказатель [4].

Метод градиентного бустинга позволяет оптимизировать произвольную дифференцируемую функцию потерь. Данный алгоритм похож на метод градиентного спуска, применяемый для решения задач оптимизации. Основная идея заключается в том, что каждый следующий добавляемый в композицию алгоритм настраивается на остатки предыдущих алгоритмов.

Пусть дана дифференцируемая функция потерь $L(F(x), y)$, где $F(x)$ – предсказанные моделью значения и y – исходные значения целевой переменной. Сам алгоритм выглядит следующим образом:

1) инициализируем композицию константным значением $F_0(x) = \arg\min \sum_{i=1}^{n} L(\alpha, y_i)$;

2) для всех $t = \overline{1,T}$, где T – итоговое количество базовых алгоритмов в композиции:

а) так как функция многих переменных максимально убывает в направлении своего антиградиента, вычисляем остатки предыдущей композиции:

$$r_{it} = -\left[\nabla_{F(x)} L(F(x_i), y_i)\right]_{F(x)=F_{t-1}(x)}, \; i = \overline{1,n};$$

б) настраиваем базовый алгоритм $b_t(x)$ на полученные остатки, то есть обучаем его по выборке $\{(x_t, r_{it}), i = \overline{1,n}\}$;

в) вычисляем коэффициент α_t перед базовым алгоритмом $b_t(x)$ путем решения следующей одномерной задачи оптимизации: $a_t = \arg\min \sum_{i=1}^{n} L(F_{t-1}(x_i) + \alpha b_t(x_i), y_i)$;

г) добавляем полученное слагаемое в композицию: $F_t(x) = F_{t-1}(x) + \alpha_t b_t(x)$.

Итогом алгоритма будет композиция $F_t(x)$ из базовых моделей $b_t(x)$ скорректированных некоторым вектором α_t, с помощью которого функция потерь $L(f(x), y)$ минимизируется. В качестве базовых моделей используются деревья решений. В качестве функции потерь будем использовать среднюю квадратичную ошибку.

Реализация наиболее эффективных моделей машинного обучения была достигнута с помощью формирования дополнительных признаков, предположительно влияющих на целевую переменную. Помимо признаков, добавленных на этапе первичного анализа, выборка была дополнена средними показателями продаж по категориальным признакам, таких как «День недели», «День месяца», «Наличие описания продукта», «Наличие изображения продукта», «Длина заголовка объявления», «Количество слов в заголовке», «Длина описания объявления», «Количество слов в описании».

Для исследования важности признаков и построения качественной модели прогнозирования были реализованы XGBoost, CatBoost и LightGBM модели машинного обучения на основе градиентного бустинга деревьев решений. Гиперпараметры к каждой модели подбирались с помощью метода кросс-валидации [5].

Оценка точности модели проводилась с помощью средней квадратичной ошибки прогнозирования (RMSE – Root Mean Squared Error) – это корень от квадрата ошибки.

$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}$, где n – количество объектов, y_t – фактическое значение целевой переменной объекта t, $\hat{y}_t$ – предсказанное значение целевой переменной объекта t.

Данную метрику легко интерпретировать, поскольку она имеет те же единицы измерения, что и целевая переменная исходного набора данных. Также она оперирует меньшими величинами по абсолютному значению, что может быть полезно для вычисления на компьютере.

Результаты построенных моделей приведены в табл. 3. Наименьшую ошибку показала модель CatBoost.

Таблица 3

Точность построенных моделей

Модель	**Показатель RMSE**
XGBoost	0.230
LightGBM	0.239
CatBoost	0.222

3. Оценка важности признаков

Для анализа результатов модели и определения значимых признаков была выбрана модель, достигшая наилучших результатов в точности прогнозирования – CatBoost, и библиотека SHAP.

SHAP (SHapley Additive exPlanations) – это теоретико-игровой подход для объяснения результатов любой модели машинного обучения. Данные метод помогает разбить прогноз, чтобы показать влияние каждой функции. Он основан на векторе Шепли – методе, который используется в теории игр для определения того, насколько каждый игрок в совместной игре внес свой вклад в ее успех [6].

В качестве примера рассмотрим влияние признаков на целевую переменную для некоторого произвольно выбранного объявления из тестовой выборки с помощью функции SHAP waterfall (рис. 5). Значения важности SHAP показывают, как сильно каждый из признаков повлиял на результат прогноза целевой переменной в сравнении с базовым значением целевой переменной (средним значением по всей обучающей выборке).

Конкретно для данного объявления наибольшее отрицательное влияние оказывают признаки «Цена» и «Рейтинг продаж по 1-му параметру», они изменяют прогноз целевой переменной на $-0,05$ и $-0,03$ соответственно. Положительно на целевую переменную для данного объявления влияют признаки «Параметр 1» и «Подкатегория продукта». В совокупности влияний всех признаков вероятность продажи по данному объявлению уменьшается до значения 0.107 по сравнению с базовым значением (средним значением по всей обучающей выборке), равным 0.139.

Для оценки общего влияния признаков на целевую переменную в рамках всей совокупности данных был построен сводный график, который показывает, какие функции наиболее важны, а также диапазон и характер их влияния на набор данных (рис. 6).

По вертикальной оси Y перечислены признаки исходя из их силы влияния на результат (в порядке убывания). Каждая точка представляет предсказанное значение целевой переменной для одного объявления. Если фактическое значение в наборе данных было высоким, цвет будет красным. Синий указывает на низкое фактическое значение. Серый цвет представляет категориальные значения, которые не могут быть маленькими или большими. Горизонтальное расположение показывает, вызвало ли влияние этого значения более высокий или более низкий прогноз [7].

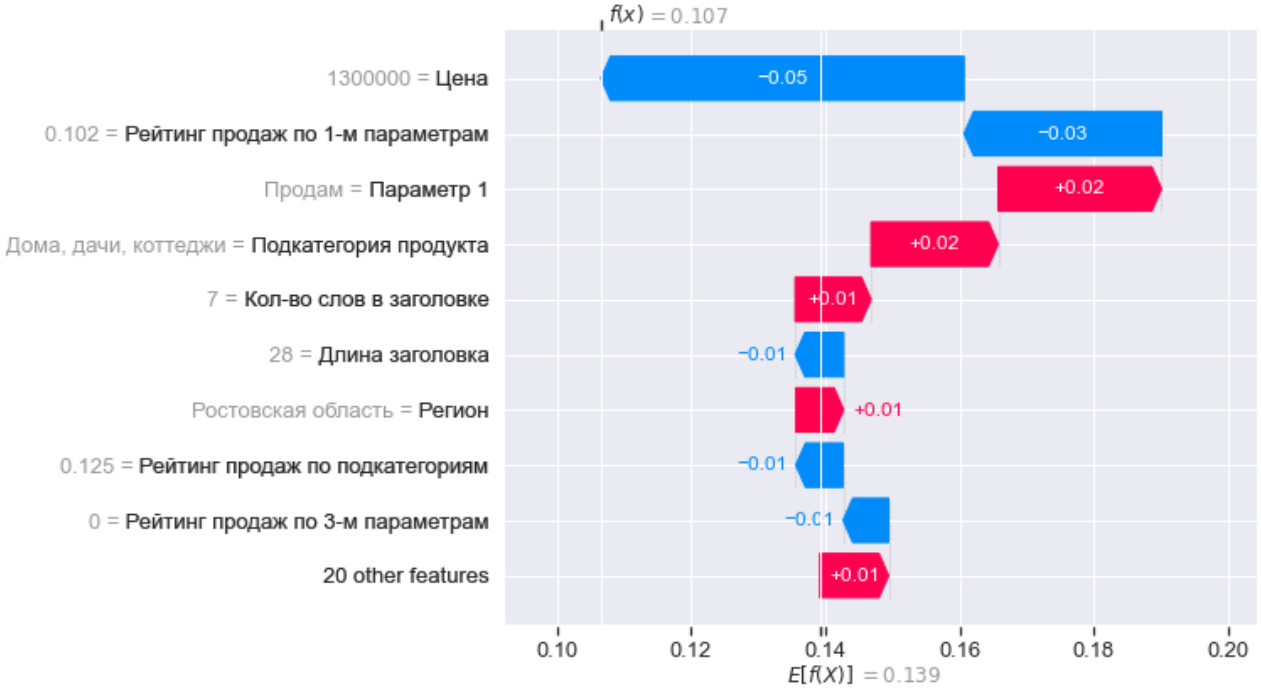


Рис. 5. Влияние признаков на целевую переменную для конкретного объявления

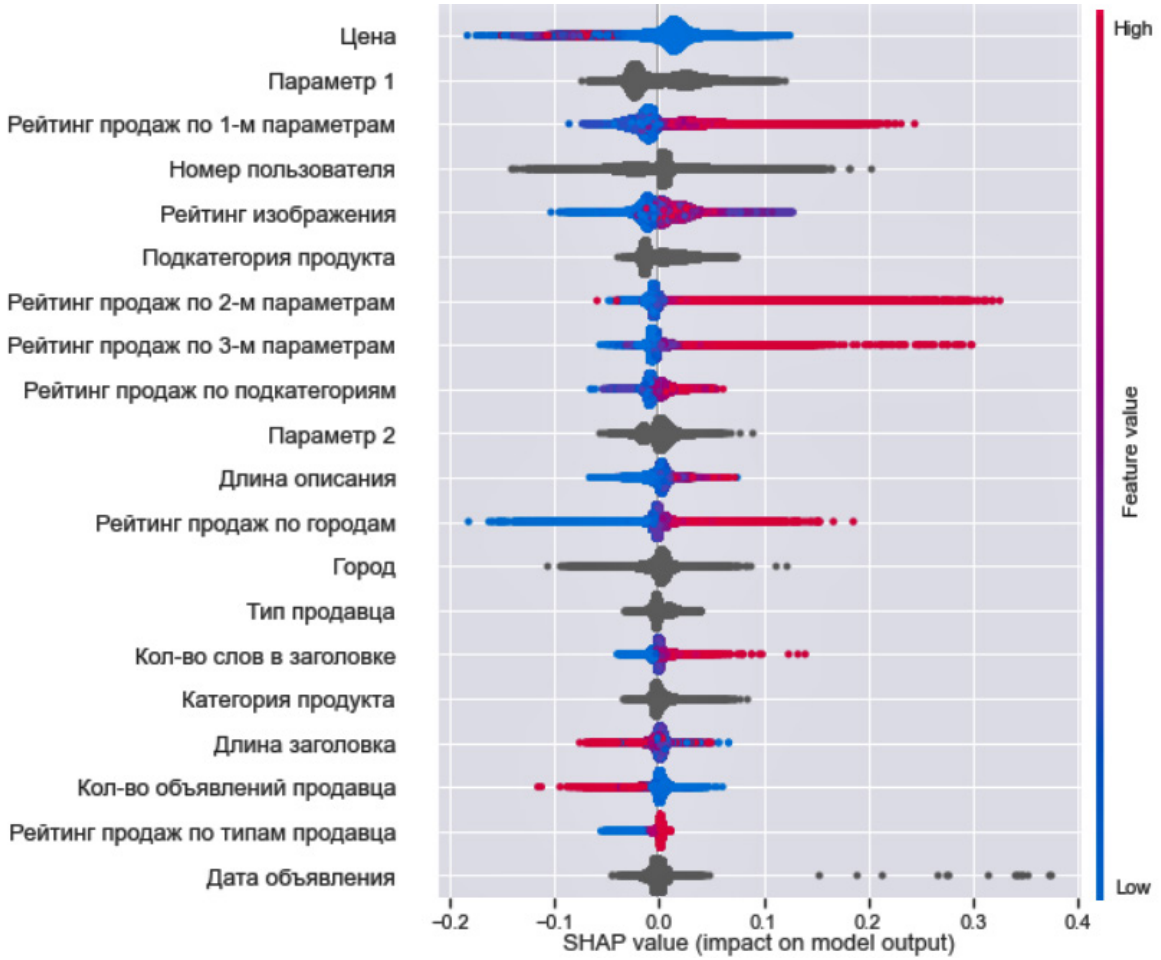


Рис. 6. Сводный график влияния признаков

На основании данного графика можно сделать выводы, что низкая цена (за редким исключением) положительно сказывается на целевой переменной, высокие рейтинги продаж по всем параметрам оказывают положительное влияние на вероятность сделки. Низкий рейтинг изображения, высокое количество объявлений у продавца, маленькая длина описания объявления и малое количество слов в заголовке понижает шанс продажи продукта.

Заключение

В ходе данного исследования были проведены анализ и обработка данных, сформированы дополнительные признаки, построено несколько моделей машинного обучения на основе градиентного бустинга деревьев решений для прогнозирования спроса, определена наиболее эффективная модель и проведен анализ ее результатов с целью поиска наиболее значимых признаков и оценки характера их влияния.

В итоговой табл. 4 выделены признаки, наиболее сильно влияющие на целевую переменную. Они разделены на уже имеющиеся в исходной выборке и новые, сформированные в процессе исследовательской работы.

Это наиболее полезные признаки, они помогают оценить эффективность описания продукта в объявлении и ответить на вопрос: какие факторы влияют на статистику продаж и каким образом?

С помощью качественной обработки данных, выделения новых признаков, подбора гиперпараметров модели с помощью кросс-валидации была построена модель CatBoost с наилучшей в рамках исследования точностью RMSE = 0.222.

Таблица 4

Значимые признаки

Тип признаков	Переменные
Имеющиеся в исходной выборке	Цена, параметр 1, номер пользователя, рейтинг изображения, подкатегория продукта, параметр 2, город, тип продавца, категория продукта, количество объявлений продавца, дата объявления
Сформированные	Рейтинг продаж по 1-м параметрам, рейтинг продаж по 2-м параметрам, рейтинг продаж по 3-м параметрам, рейтинг продаж по подкатегориям, длина описания объявления, рейтинг продаж по городам, количество слов в заголовке, длина заголовка объявления, рейтинг продаж по типам продавца

Литература

1. Avito Demand Prediction Challenge. – Available at: https://www.kaggle.com/c/avito-demand-prediction (accessed 15 March 2021)

2. *Плас, Дж. В.* Python для сложных задач: наука о данных и машинное обучение / Дж. В. Плас. Пер. с англ. – Санкт-Петербург : Питер, 2018. – 576 с.

3. *Жерон, О.* Прикладное машинное обучение с помощью Scikit-learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем / О. Жерон. Пер. с англ. – Севастополь : O'Reilly Media, 2018. – 688 с.

4. *Мюллер, А.* Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными / А. Мюллер, С. Гвидо. Пер. с англ. – Москва : Альпина Паблишерз, 2017. – 393 с.

5. *Шолле, Ф.* Глубокое обучение на Python / Ф. Шолле. Пер. с англ. – Санкт-Петербург : Питер, 2018. – 400 с.

6. Interpretable Machine Learning. – Available at: https://towardsdatascience.com/interpretable-machine-learning-1dec0f2f3e6b (accessed 28 March 2021)

7. Tutorial: Explainable Machine Learning with Python and SHAP. – Available at: https://mlconference.ai/blog/tutorial-explainable-machine-learning-with-python-and-shap (accessed 28 March 2021)

Петрова Дарья Александровна – магистрант 1-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: darya_petrova@mail.ru

Каширина Ирина Леонидовна (научный руководитель) – д-р техн. наук, проф., профессор кафедры математических методов исследования операций Воронежского государственного университета. E-mail: kash.irina@mail.ru

УДК 004.942

ИССЛЕДОВАНИЕ РЕЖИМОВ ПЕРЕДАЧИ ДАННЫХ В КОМПЬЮТЕРНЫХ СЕТЯХ

А. Н. Пешкова

Воронежский государственный университет

Введение

При проектировании и модернизации вычислительных сетей основной задачей является определение параметров сети для ее эффективного функционирования. Для этого используют методы математического моделирования.

Механизм разделения среды протокола Ethernet, наиболее часто используемого в компьютерных сетях, упрощенно описывается одноканальной моделью с простейшим потоком заявок и показательным законом распределения времени обслуживания (модель типа M/M/1). Она хорошо описывает процесс обработки поступающих заявок на обслуживание системами с одним обслуживающим прибором со случайным временем обслуживания. При генерации пакета на сетевом устройстве он помещается в буфер для хранения на время, пока обслуживающий прибор занят выполнением другой заявки. После передачи пакета, ожидается подтверждение его доставки, после чего пакет удаляется из буфера. Если время ожидания подтверждения превышает установленное, пакет отправляется повторно. Передающая среда Ethernet представлена в этой модели обслуживающим устройством, а пакеты соответствуют заявкам [2].

Качество функционирования сети с использование технологий Ethernet можно определять загруженностью сети, временем передачи информации, наличием и количеством потерянных пакетов. Для решения этой задачи необходимо построение математической модели и проведения ее анализа.

1. Построение и анализ модели

Для построения математической модели необходимо построить граф состояний системы. При этом были приняты следующие допущения:

• не учитывается установление соединения, так как оно не влияет на время передачи пакетов потому, что происходит только один раз до отправки;

• время прохождения пакета и время подтверждения считаем одним событием, так как они зависят от загрузки канала и интенсивности отправки передаваемых пакетов [1].

Существенное влияние на сложность модели будет оказывать количество пакетов, одновременно находящихся в системе. Для анализа рассмотрены режимы, при которых одновременно на устройстве находится не более трех пакетов (рис. 1). Каждое состояние характеризуется тремя значениями: первое – количество пакетов в буфере, ждущих очередь на отправку; второе – количество отправленных пакетов, ждущих подтверждения доставки; третье – флаг подтверждения, который может принимать одно из трех значений: 0 – подтверждения нет, 1 – подтверждение пришло, no – пакет не доставлен.

Интенсивности перехода, представленные на графе, распределены по экспоненциальному закону. При начальных условиях:

$$P_{0,0,0}(t) = 0; P_{1,0,0}(t) = P_{1,1,0}(t) = P_{1,1,1}(t) = P_{2,0,0}(t) = P_{2,1,0}(t) = P_{2,1,1}(t) = P_{2,2,0}(t) = P_{2,2,1}(t) = P_{3,0,0}(t) =$$
$$= P_{3,1,0}(t) = P_{3,1,1}(t) = P_{3,2,0}(t) = P_{3,2,1}(t) = P_{3,3,0}(t) = P_{3,3,1}(t) = 0$$

уравнения Колмогорова имеют вид:

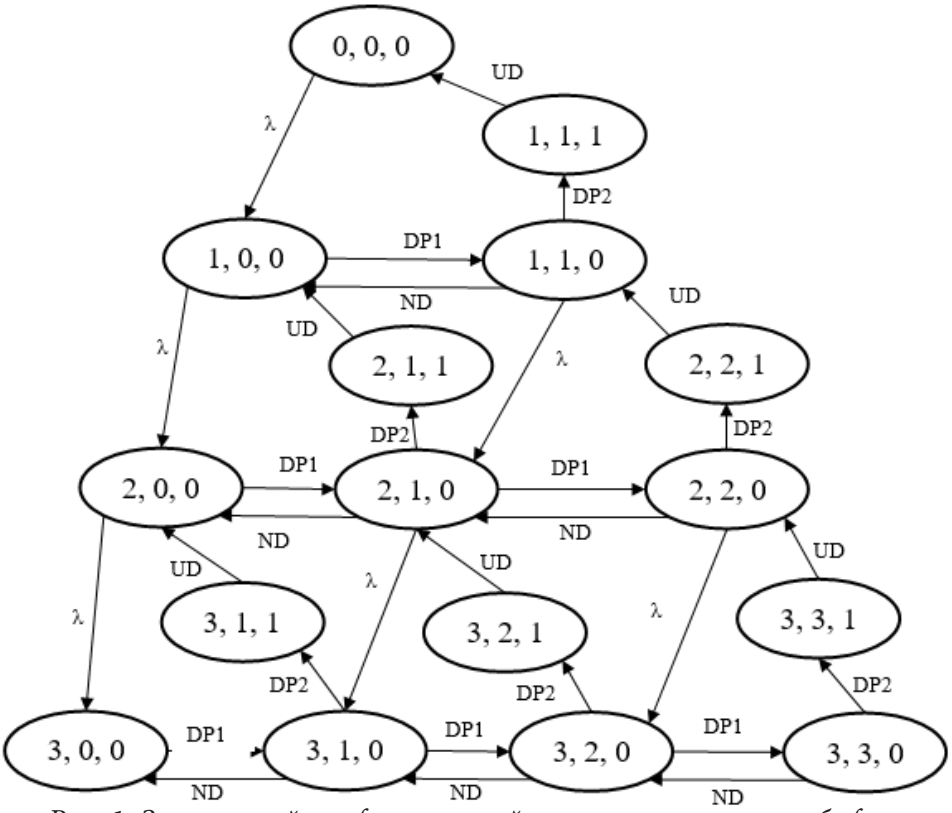


Рис. 1. Замкнутый граф состояний с тремя пакетами в буфере

$$
\begin{cases}
\dfrac{d}{dt}P_{0,0,0}(t) = -\lambda \cdot P_{0,0,0}(t) + UD \cdot P_{1,1,1}(t) \\
\dfrac{d}{dt}P_{i,0,0}(t) = \lambda \cdot P_{i-1,0,0}(t) - \alpha_2 \cdot \lambda \cdot P_{i,0,0}(t) - DP1 \cdot P_{i,0,0}(t) + ND \cdot P_{i,1,0}(t) + \alpha_2 \cdot UD \cdot P_{i+1,1,1}(t), i = 1,2,3 \\
\dfrac{d}{dt}P_{i,j,0}(t) = DP1 \cdot P_{i,j-1,0}(t) + \alpha_2 \cdot ND \cdot P_{i,j+1,0}(t) - j \cdot DP2 \cdot P_{i,j,0}(t) - \alpha_2 \cdot DP1 \cdot P_{i,j,0}(t) - \\
-ND \cdot P_{i,j,0}(t) - \alpha_4 \cdot \lambda \cdot P_{i,j,0}(t) + \alpha_5 \cdot \lambda \cdot P_{i-1,j,0}(t) + \alpha_4 \cdot UD \cdot P_{i+1,j+1,1}(t), i = 1,2,3, j = 1..i \\
\dfrac{d}{dt}P_{i,j,1}(t) = j \cdot DP2 \cdot P_{i,j,0}(t) - UD \cdot P_{i,j,1}(t), j = 1,2,3
\end{cases}
$$

где λ – интенсивность поступления пакетов для передачи, $DP1$ – интенсивность отправки пакета, которая зависит от размера передаваемого пакета и параметров передающего устройства, $DP2$ – интенсивность доставки пакета – величина обратная времени, необходимому на доставку пакета и подтверждения о доставке, она зависит от текущего протокола передачи и характеристик среды передачи. UD – интенсивность удаления пакетов из очереди, зависящая от текущего протокола передачи данных. ND – интенсивность отсутствия подтверждения за определенный интервал времени – время ожидания повторной отправки.

Для проведения анализа количества обрабатываемых пакетов в системе передачи было проведено суммирование вероятностей состояний, в которых общее количество пакетов равно 1, 2 и 3. Так для приведенного примера (рис. 2 для трех пакетов в сети) наиболее вероятное состояние – три пакета. При этом вероятность такого состояния достаточно велика (почти 0,6), что говорит о необходимости для исследования повысить сложность модели и увеличить максимальное количества пакетов в сети.

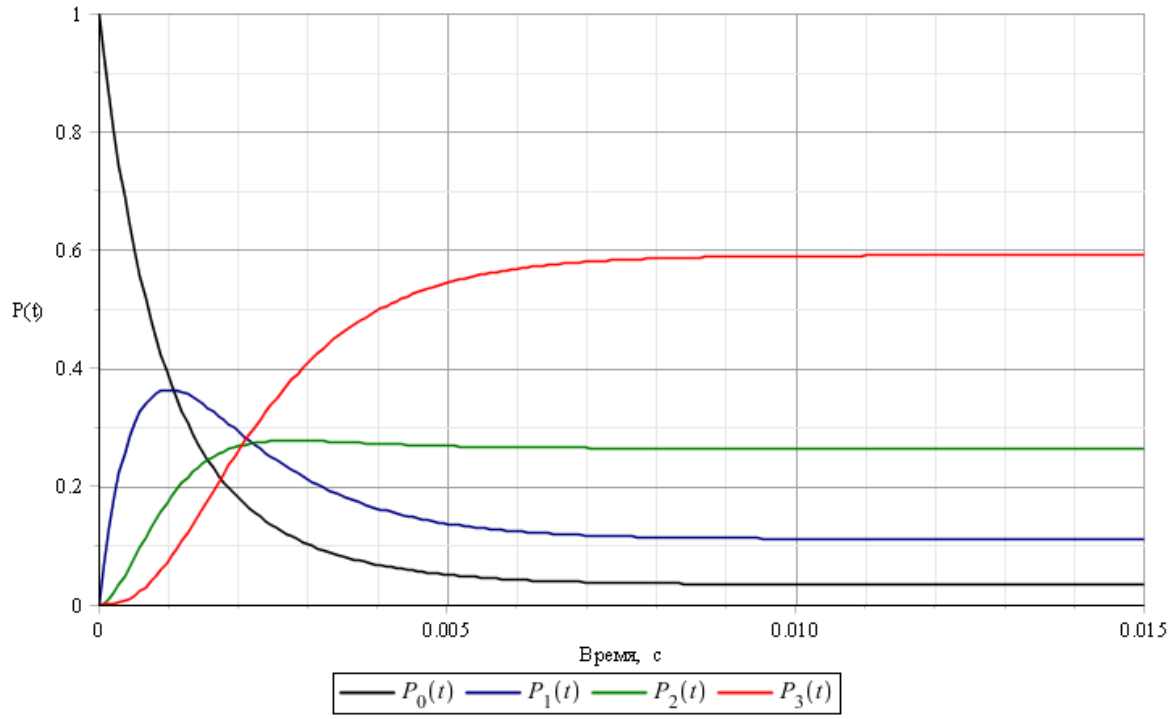


Рис. 2. Вероятность нахождения системы с определенным количеством пакетов

Результаты проведенного исследования (рис. 3) показывают зависимость наиболее вероятного состояния длины очереди от интенсивности отправки пакетов в сеть и задержках в сети. Данные результаты также иллюстрируют область применимости модели с количеством пакетов в системе до трех. При наиболее вероятной длине очереди 3 для анализа режимов функционирования сети необходимо применять более сложную модель. Так, например, при передаче до 700 пакетов в секунду и времени задержки менее 0,0001 секунды, загруженность сети небольшая (наиболее вероятная длина очереди равна 1) и можно использовать более простую модель с двумя пакетами в системе.

с					
0,001	1	2	2	3	3
0,0001	0	1	1	2	2
0,00001	0	0	1	1	2
0,000001	0	0	1	1	2
0	250	500	700	1000	1500 п/с

Рис. 3. Зависимость наиболее вероятного состояния длины очереди от интенсивности отправки пакетов и задержках времени

Одной из главных характеристик систем реального времени является время доставки пакетов. Для того, что получить эту характеристику, необходимо преобразовать систему так, чтобы состояния доставки сделать концевыми (рис.4). После чего по математическому ожиданию плотности распределения можно вычислить среднее время доставки.

На основе графа с поглощающими состояниями получаем систему уравнений Колмогорова для трех пакетов в устройстве:

$$\begin{cases} \frac{d}{dt}P_{0,0,0}(t) = -\lambda \cdot P_{0,0,0}(t) \\ \frac{d}{dt}P_{i,0,0}(t) = \lambda \cdot P_{i-1,0,0}(t) - \alpha_2 \cdot \lambda \cdot P_{i,0,0}(t) - DP1 \cdot P_{i,0,0}(t) + ND \cdot P_{i,1,0}(t), i = 1,2,3 \\ \frac{d}{dt}P_{i,j,0}(t) = DP1 \cdot P_{i,j-1,0}(t) + \alpha_2 \cdot ND \cdot P_{i,j+1,0}(t) - j \cdot DP2 \cdot P_{i,j,0}(t) - \alpha_2 \cdot DP1 \cdot P_{i,j,0}(t) - \\ -ND \cdot P_{i,j,0}(t) - \alpha_4 \cdot \lambda \cdot P_{i,j,0}(t) + \alpha_5 \cdot \lambda \cdot P_{i-1,j,0}(t), i = 1,2,3, j = 1..i \\ \frac{d}{dt}P_{i,j,1}(t) = j \cdot DP2 \cdot P_{i,j,0}(t), j = 1,2,3 \end{cases}$$

При следующих начальных условиях:

$$P_{0,0,0}(t) = 0; P_{1,0,0}(t) = P_{1,1,0}(t) = P_{1,1,1}(t) = P_{2,0,0}(t) = P_{2,1,0}(t) = P_{2,1,1}(t) = P_{2,2,0}(t) = P_{2,2,1}(t) = P_{3,0,0}(t) =$$
$$= P_{3,1,0}(t) = P_{3,1,1}(t) = P_{3,2,0}(t) = P_{3,2,1}(t) = P_{3,3,0}(t) = P_{3,3,1}(t) = 0$$

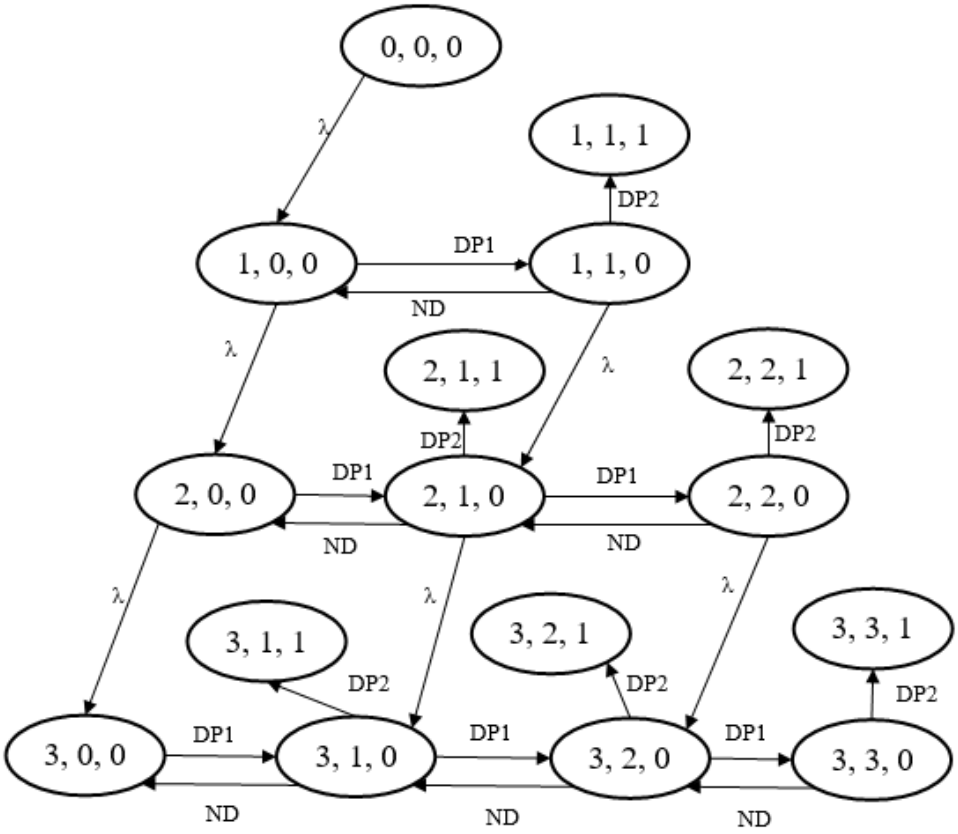


Рис. 4. Граф с поглощающими состояниями – пакет доставлен

Для данного графа была проанализирована зависимость математического ожидания от интенсивности отправки пакетов при различном времени его задержки в сети (tz) и при нахождении разного количества пакетов в буфере (n). Размер пакета фиксированный и равен 1500 байт, скорости передачи данных 100 Мбит/с. Из графика (рис. 5) видно, что математическое ожидание при одинаковом значении задержки времени в сети совпадает или различается незначительно для двух и трех пакетов в буфере, что свидетельствует о достаточности использования более простой модели с двумя пакетами при заданных условиях.

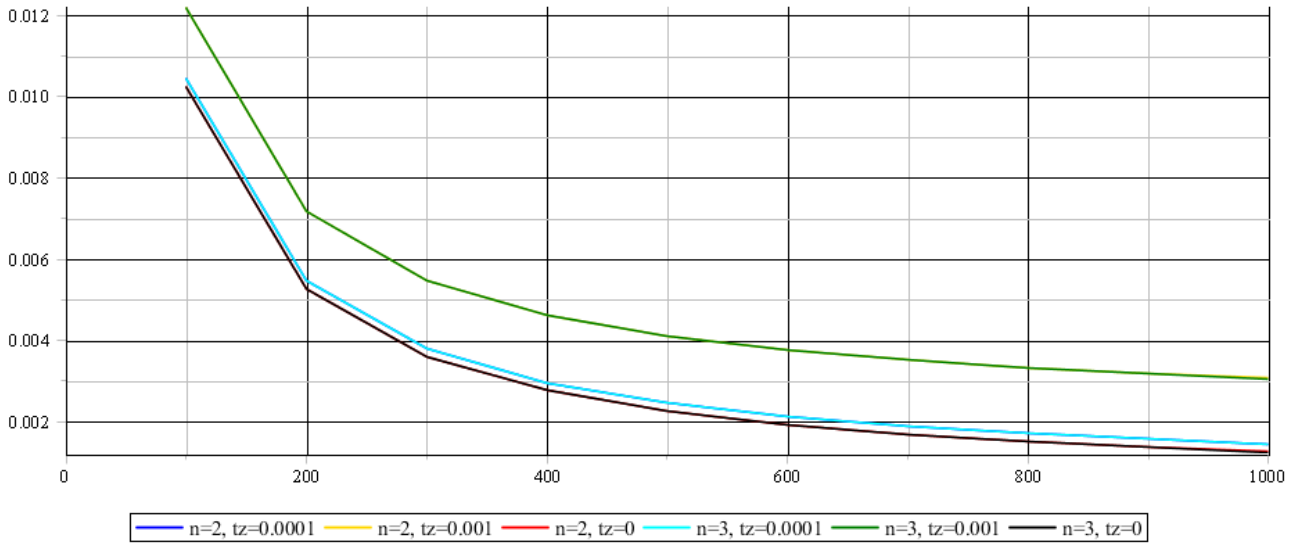


Рис. 5. Зависимость математического ожидания от интенсивности отправки пакетов

Значение математического ожидания при различных значениях интенсивности отправки пакетов приведены в табл. 1. При сравнении с экспериментальными данными можно удостовериться в адекватности разработанной модели. Полученные значения точны до 3 порядка включительно.

Таблица 1

Значение математического ожидания при различной интенсивности отправки пакетов

Интенсивность отправки пакетов	Математическое ожидание из эксперимента	Математическое ожидание из модели
100	0.01331179758	0.0133543228628815
200	0.005025073947	0.00526040196044915
500	0.003822818417	0.00383049504397128

Заключение

Таким образом, проведенный анализ показал, что предложенные подходы позволяют оценить вероятности нахождения системы передачи данных в различных состояниях в различные моменты времени для систем с заданными характеристиками, а также оценить сложность используемых моделей.

Кроме того, исследования этой модели показали, что с ее помощью можно адекватно оценить время доставки для систем с заданными характеристиками. Результаты исследований могут быть применены при моделировании систем реального времени с протоколами гарантированной доставки на основе Ethernet и настройке параметров их функционирования.

Литература

1. *Абрамов, Г. В.* Исследование передачи данных по каналу множественно доступа с гарантированной доставкой в системах реального времени / Г. В. Абрамов, В. В. Денисенко // Вестник Воронежского государственного технического университета. – 2015 – № 4 – С. 38–43.

2. *Боровков, А. А.* Вероятностные процессы в теории массового обслуживания / А. А. Боровков. – Москва : Главная редакция физико-математической литературы издательства «Наука», 2017. – 368 с.

Пешкова Анастасия Николаевна – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: pessshkova19@gmail.com

Абрамов Геннадий Владимирович (научный руководитель) – д-р тех. наук, заведующий кафедрой математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: agwl@yandex.ru

УДК 004.932

ИССЛЕДОВАНИЕ И РЕАЛИЗАЦИЯ АЛГОРИТМА ELA С ЦЕЛЬЮ ВЫЯВЛЕНИЯ МОНТАЖА НА ЦИФРОВЫХ ФОТОГРАФИЯХ

А. А. Пивоваров

Воронежский государственный университет

Введение

В данной статье описывается разработка приложения, в котором реализуется алгоритм Error Level Analysis (ELA). В первой части статьи рассматривается принцип работы алгоритма ELA, затем описывается интерфейс приложения и этапы его реализации, а потом приводятся результаты различных экспериментов с использованием оригинальных и модифицированных изображений.

1. Алгоритм поиска монтажа на фотографиях

В настоящее время существует множество программ для обработки изображений. Такие программы, как Adobe Photoshop и Gimp, позволяют не только изменять цветовые характеристики изображений, но и совмещать части нескольких изображений, удалять части изображений и применять различные локальные эффекты к изображениям.

Одним из самых популярных форматов изображений в настоящее время является формат JPEG. Большинство фотографий сохраняются в этом формате.

1.1. Обзор сжатия изображений JPEG

Формат JPEG использует сжатие с потерями и основан на сочетании пространственной и частотной области. Исходное изображение делится на блоки 8×8, и каждый блок преобразуется в область частот с использованием дискретного косинусного преобразования (DCT). Коэффициенты DCT упорядочиваются зигзагообразно и квантуются с использованием таблицы квантования. Кодирование без потерь сжимает коэффициенты квантования, чтобы сформировать файл формата JPEG. Обратный процесс происходит при распаковке файла формата JPEG. Размер конечного изображения зависит от содержания и качества изображения. Однородное изображение с небольшим количеством объектов требует меньшего количества бит по сравнению со сложным изображением с множеством объектов. Многие манипуляции с изображениями с помощью Adobe Photoshop или GIMP обычно повторно сохраняют и сжимают его как новое изображение формата JPEG.

1.2. Алгоритм Error Level Analysis

Алгоритм ELA – это один из методов обнаружения манипуляций с изображением путем повторного сохранения изображения с определенным уровнем качества и последующего вычисления разницы между уровнями сжатия [2–3]. Если над изображением не совершались манипуляции, то квадраты 8×8 должны иметь одинаковые потенциалы ошибок. Однако, если изображение изменено, часть изображения, которая была изменена, должна иметь более высокий потенциал ошибки, чем другая часть изображения. Алгоритм ELA намеренно сохраняет изображение с известным коэффициентом ошибок, например 95 %, а затем вычисляет разницу между исходным и полученным изображениями.

При первом сохранении фотографии в формате JPEG она впервые сжимается. Большинство программ для редактирования изображений, таких как Adobe Photoshop или GIMP, поддерживают сжатие JPEG. Следовательно, если изображение затем открыть в редакторе, изменить и снова сохранить в формате JPEG, оно будет сжато еще раз. После этого исходные части фотографического изображения будут сжаты дважды: один раз камерой, сделавшей фотографию, и еще раз при сохранении в редакторе, в то время как «отредактированная» часть фотоизображения была сжата в редакторе только один раз. Человек не сможет отличить два изображения на глаз. Зато мы можем сравнить два изображения с помощью программы и увидеть разницу более наглядно. На рис. 1 показан пример таблиц квантования для светимости (Y) и цветости (CrCb) [1]. С помощью этих таблиц можно вычислить примерное значение качества изображения JPEG. Для этого можно подставить средние значения светимости и цветности в формулу:

$$u = \frac{Y + Cr + Cb}{3},$$

$$\Delta = |Y - Cr| \cdot 0.5 + |Y - Cb| \cdot 0.5,$$

$$Q = 100 - u - \Delta.$$

На рис. 2 представлен пример значений уровня ошибки, вычисленных при помощи алгоритма ELA. Значения, которые отличаются от остальных, помогают определить область, над которой проводились манипуляции.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	58	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

светимость

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

цветность

Рис. 1. Таблицы квантования

81	81	81	82	82	81	81	81
81	81	81	82	82	81	81	81
81	81	81	81	81	81	81	81
91	91	91	91	81	81	81	81
91	91	91	91	81	81	81	81
91	91	91	91	81	81	80	81
81	81	81	81	81	81	80	81
81	81	82	82	81	81	81	81
81	81	81	81	81	81	80	80
81	81	81	81	81	81	80	80

Рис. 2. Уровень ошибки

2. Программная реализация

Приложение реализовано на языке программирования C#. Для реализации интерфейса использовалась технология Windows Forms. Для разработки приложения использовалась среда разработки Visual Studio.

2.1. Основной алгоритм

Упрощеная версия основного алгоритма программы выглядит следующим образом:

1. Загружается исходное изображение.

2. Выбираются значения трех параметров: качества изображения, количества пересохранений и коэффициента умножения ошибки. Параметр качества отвечает за то, с каким качеством будет пересохраняться изображение. Параметр количества пересохранений отвечает за то, сколько раз изображение будет пересохранено перед сравнением. Параметр коэффициента умножения ошибки отвечает за то, насколько яркими будут области с высоким значением ошибки в результате.

3. Изображение пересохраняется выбранное количество раз с выбранным качеством.

4. Исходное и пересохраненное изображение попиксельно сравниваются.

5. Результирующее изображение представляет из себя разницу между исходным и пересохраненным изображением, которая искуственно поднята с помощью коэффициента умножения ошибки.

2.2. Пользовательский интерфейс

Основное окно приложения представленно на рис. 3. Меню File позволяет загрузить произвольное изображение в программу. Меню ELA позволяет применить алгоритм ELA к загруженному изображению. Параметр Quality отвечает за качество пересохраняемого изображения. Error Multiplier – коэффициент умножения ошибки. Re-saves – количество пересохранений перед сравнением.

Основное окно также содержит две вкладки с изображением. В первой можно увидеть оригинальное изображение, во второй – результат работы алгоритма.

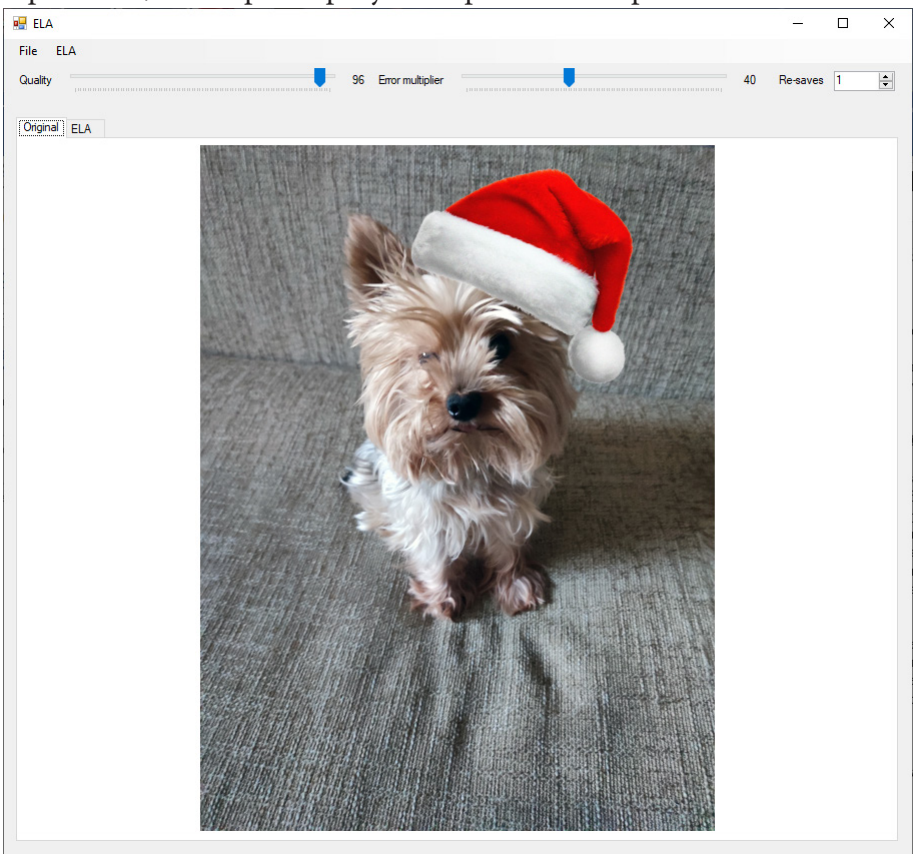


Рис. 3. Основное окно приложения

3. Анализ изображений с помощью приложения

В данном разделе будут приведены примеры фотографий с применением монтажа и анализ результатов работы программы. Для демонстрации независимости работы алгоритма от типа устройства фотографии были сделаны на смартфон и на отдельный фотоаппарат. Для демонстрации независимости работы алгоритма от выбора графического редактора фотографии были обработаны в двух различных графических редакторах: Adobe Photoshop и GIMP. Параметры были подобраны экспериментальным путем для лучшей презентации алгоритма.

3.1. Монтаж с использованием стороннего источника

Для примера возьмем фотографию собаки, сделанную с помощью смартфона. В графическом редакторе GIMP добавим собаке шапочку, оригинал которой был найден в интернете. Конечное изображение со качеством 95 % экспортируем в формат JPEG и загрузим в программу. Параметры зададим следующим образом: Quality: 95 %, Error multiplier: 40, Re-saves: 1. Исходное изображение и результат можно увидеть на рис. 4.

Рис. 4. Результат работы программы

В данном случае явно видно, что уровень ошибки в районе «шапочки» сильно отличается от остальных областей, что является признаком монтажа. Это объясняется в первую очередь тем, что изображение шапочки изначально было в формате PNG, поэтому первый раз оно было сжато в графическом редакторе, в отличие от изображения собаки, которое первый раз было сжато на смартфоне.

3.2. Монтаж с использованием второй фотографии

Для примера возьмем две фотографии синиц, сделанные с помощью фотоаппарата. В графическом редакторе Photoshop мы возьмем синицу с одной фотографии и добавим ее в другую. Конечное изображение с качеством 95 % экспортируем в формат JPEG и загрузим в про-

грамму. Параметры зададим следующим образом: Quality: 88 %, Error multiplier: 60, Re-saves: 2. Исходное изображение и результат можно увидеть на рис. 5.

В данном случае область монтажа не так ярко выражена, так как обе фотографии взяты из одного источника. Можно заметить, что выделяются области в районе головы левой синицы, а также области в районе веток. Однако это не является результатом монтажа, а лишь показывает наличие ярких однородных областей (в данном случае белых). На это следует обращать пристальное внимание при визуальном анализе. С другой стороны, несмотря на то что обе синицы на фотографии выглядят примерно одинаково, левая сливается с фоном в результирующем изображении, тогда как правая выделяется более темной областью. Такие различия могут свидетельствовать о наличии монтажа.

Рис. 5. Результат работы программы

Литература

1. *Сэломон, Д.* Сжатие данных, изображений и звука / Д. Сэломон. – Москва : Техносфера, 2004. – 368 с.

2. Error level analysis – Режим доступа: https://en.wikipedia.org/wiki/Error_level_analysis.

3. Photo Forensics: Detect Photoshop Manipulation with Error Level Analysis – Режим доступа: https://resources.infosecinstitute.com/topic/error-level-analysis-detect-image-manipulati on.

4. JPEG Compression, Quality and File Size – Режим доступа: https://www.impulseadventure.com/photo/jpeg-compression.html.

5. EMGU – Working with images – Режим доступа: https://www.emgu.com/ wiki/index.php/Working_with_Images.

Пивоваров Артем Андреевич – магистрант 2-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: rainkear@ktbk.ru

Корольков Олег Геннадьевич (научный руководитель) – канд. физ.-мат. наук, доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: korolkov@amm.vsu.ru

УДК 004.42

РЕШЕНИЕ ЗАДАЧИ ОБ ОПТИМАЛЬНОЙ УПАКОВКЕ В КОНТЕЙНЕРЫ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Н. В. Покатаев

Воронежский государственный университет

Введение

Настоящая работа посвящена применению генетических алгоритмов к решению задачи об оптимальной упаковке в контейнеры, имеющей многочисленное число приложений в различных отраслях промышленности. В задачах двумерной прямоугольной упаковки требуется без взаимного перекрытия разместить имеющиеся прямоугольные элементы внутри некоторых плоских прямоугольных объектов (контейнеров). Предполагается, что стороны размещаемых прямоугольников должны быть параллельны сторонам контейнера. Различают задачи, в которых можно размещаемые прямоугольники поворачивать на 90о либо такой поворот запрещен. На практике задачи с запрещением поворота возникают, например, при верстке материала в полиграфическом производстве или при раскрое декорированного либо структурированного материала в мебельной, стекольной или швейной промышленности.

1. Постановка задачи

Математическая модель сформулированной задачи представляет собой задачу упаковки некоторого количества элементов в минимальное число контейнеров [1]. При этом задача может быть сформулирована следующим образом.

Дано множество контейнеров размера V и набор n прямоугольных элементов для упаковки с размерами $a_1, a_2, ..., a_n$. Найти целое число контейнеров B и B-разбиение $S_1 \bigcup ... \bigcup S_B$ множества $\{1, ..., n\}$ такое, что $\sum_{i \in S_k} a_i \le V, \ \forall k = 1, ..., B,$ и число контейнеров B было минимально.

Введем обозначения:

$$y_i = \begin{cases} 1, \text{если контейнер } i \text{ использован}, \\ 0, \text{ в противном случае}, \end{cases} \quad i = 1, ..., n \tag{1}$$

$$x_{ij} = \begin{cases} 1, \text{если элемент } j \text{ упакован в контейнер } i, \\ 0, \text{ в противном случае}, \end{cases} \tag{2}$$

$$i = 1, ..., n, \ \ j = 1, ..., n.$$

Тогда задача запишется в виде:

$$B = \sum_{i=1}^{n} y_i \to \min \tag{3}$$

при

$$B \ge 1 \tag{4}$$

$$\sum_{j=1}^{n} a_j x_{ij} \le V y_i, \forall i = 1, ..., n \tag{5}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1, ..., n \tag{6}$$

$$y_i \in \{0,1\}, i = 1,...,n \quad (7)$$

$$x_{ij} \in \{0,1\}, i = 1,...,n, j = 1,...,n \quad (8)$$

Условие (4) гарантирует ненулевое решение. Условие (5) означает, что суммарный размер деталей, упакованных в *i*-й контейнер, не превышает его размера. Условие (6) означает, что *j*-й элемент должен быть упакован только в один контейнер.

Рассмотренная задача относится к классу NP-трудных задач комбинаторной оптимизации. В качестве метода решения данной задачи будем использовать генетические алгоритмы. Данный метод широко применяется для решения задач упаковки. В данной работе рассмотрен случай одного контейнера. В дальнейшем планируется его модифицировать под случай с несколькими контейнерами.

2. Общая схема генетического алгоритма

Существует несколько возможных схем устройства генетического алгоритма. Все они отличаются друг от друга порядком следования действий, направленных на развитие популяции. Для данной задачи используется схема, представленная на рис. 1.

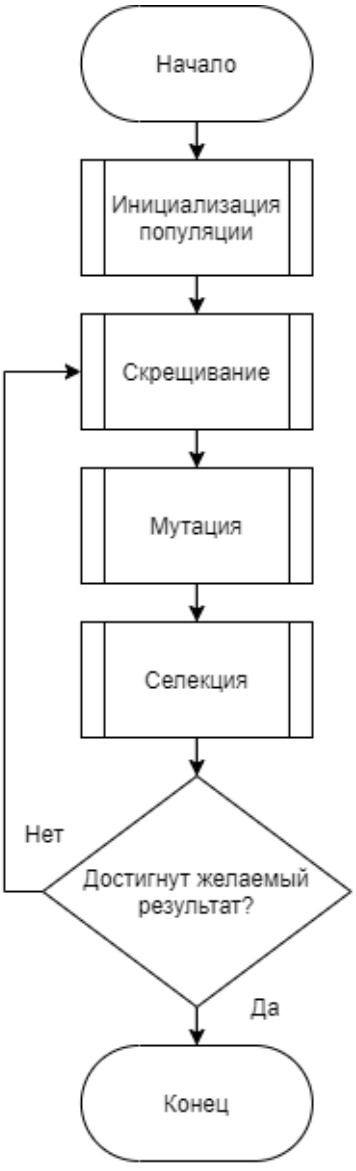


Рис. 1. Общая схема генетического алгоритма

3. Алгоритм укладки

Требуется расположить все заданные прямоугольники таким образом, чтобы расстояние от верхнего ребра самой удаленной от оси абсцисс фигуры до этой оси было минимальным.

Алгоритм укладки деталей имеет непосредственную связь со способом представления данных. Для применения генетического алгоритма в рамках конкретной задачи используется особенный вектор-генотип [2]. Способ упаковки кодируется перестановкой $\pi = (i_1, i_2, ..., i_n)$, где i_k – номер детали. Перестановка представляет собой последовательность, в которой элементы упаковываются в контейнер. Каждый элемент такого вектора является геном (порядковым номером фигуры в некотором конечном множестве прямоугольников).

Согласно этому генотипу, на этапе определения приспособленности популяции, идет построение решения:

1. Выбирается фигура с порядковым номером, находящимся в первой координате вектора-генотипа.

2. От левого нижнего края листа совершается попытка установки выбранной фигуры (рис. 2). В случае, если установка была неудачна, попытка повторяется от соседней точки справа или сверху (если справа недостаточно места для размещения фигуры данного размера (рис. 3)).

Построение продолжается до тех пор, пока все фигуры не будут установлены. Данное правило размещения способствует равномерному расположению фигур на нижних ярусах и минимизирует занятую высоту листа.

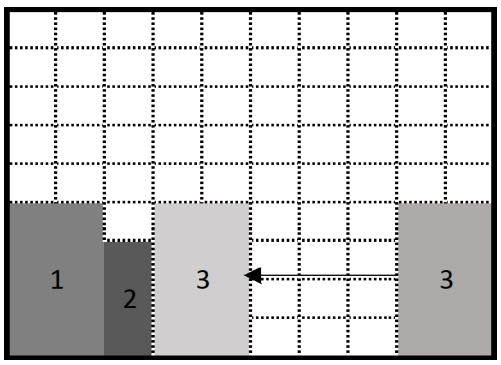


Рис. 2. Попытка расположения очередной фигуры (если справа есть доступное место)

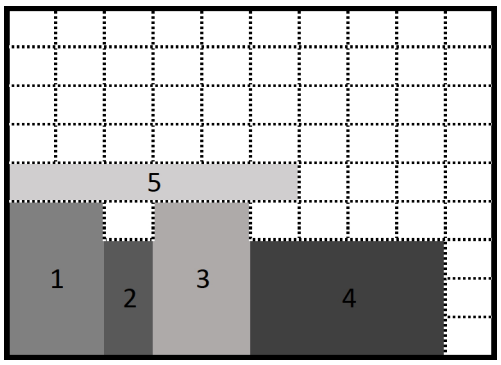


Рис. 3. Попытка расположения очередной фигуры (если справа нет доступных мест)

4. Реализация основных событий

4.1. Инициализация популяции

Алгоритм генерации начальной популяции представляет собой создание псевдослучайной перестановки на дополненном множестве натуральных чисел, ограниченном сверху порядковым номером последнего прямоугольника в заданной коллекции. Генерация генотипов происходит ровно один раз в начале первой итерации жизненного цикла популяции.

4.2. Обмен генов (кроссинговер)

Алгоритм обмена генов учитывает, что исходный набор представляет собой перестановку порядковых номеров конечного множества прямоугольников, а значит каждый элемент является уникальным. Кроссинговер проходит в несколько этапов:

1. Выбираются два соседних генотипа. Например,

$\pi_i = (1,2,3,4,5,6)$

$\pi_j = (6,4,2,5,3,1)$

2. Генерируется псевдослучайное число – место p в пределах первого генотипа. Например, $p = 3$.

3. Из первого генотипа копируются первые p позиций в новый генотип.

$$\pi_{new} = (1,2,3,...)$$

4. Из второго генотипа добавляются отсутствующие гены в конец первого.

$$\pi_{new} = (1,2,3,6,4,5)$$

Затем родители меняются местами и создается еще один потомок. В результате число особей популяции возрастает в 2 раза.

4.3. Мутации

Случайные изменения в генотипе затрагивают только потомков, созданных на текущей итерации:

1. Генерируются два псевдослучайных числа – индексы двух случайных генов в пределах генотипа.

2. Значения в заданных индексах меняются местами.

В каждом векторе происходит ровно одна мутация.

4.4. Выбор приспособленных особей (селекция)

После завершения фаз кроссинговера и мутации начинается выбор приспособленных особей. В данном процессе погибает половина особей, показавшая худший коэффициент выживаемости. Для оставшихся генотипов-потомков начинается следующий этап в жизненном цикле популяции.

Определением выживаемости генотипа занимается некоторая специальная, целочисленная целевая функция (фитнес-функция). Данная задача является одной из важнейших, так как направляет эволюцию популяции в сторону оптимального решения. От скорости выполнения сильно зависят темпы развития векторов-генотипов внутри популяции. В настоящей работе в качестве фитнес-функции рассмотрена функция

$$f : \pi \to \mathbb{R}_+$$

представляющая собой верхнюю границу высот, полученную при укладке, и обладающая свойством: $f(\pi_i) < f(\pi_j)$, если перестановка π_i представляет «лучшую» укладку, чем перестановка π_j.

5. Архитектура приложения

Архитектура приложения представлена на рис. 4. Популяция в данном экземпляре задачи представляет собой контейнер, хранящий информацию о генотипах, фигурах и свойствах листа. Основные действия, такие как кроссинговер, селекция, мутация и подсчет коэффициента выживаемости, сосредоточены в ее жизненном цикле. Данный объект определяет порядок применения событий, происходящих с популяцией. Все действия (события) реализуют поведенческий шаблон проектирования стратегия и могут быть подменены на каком-либо этапе выполнения приложения.

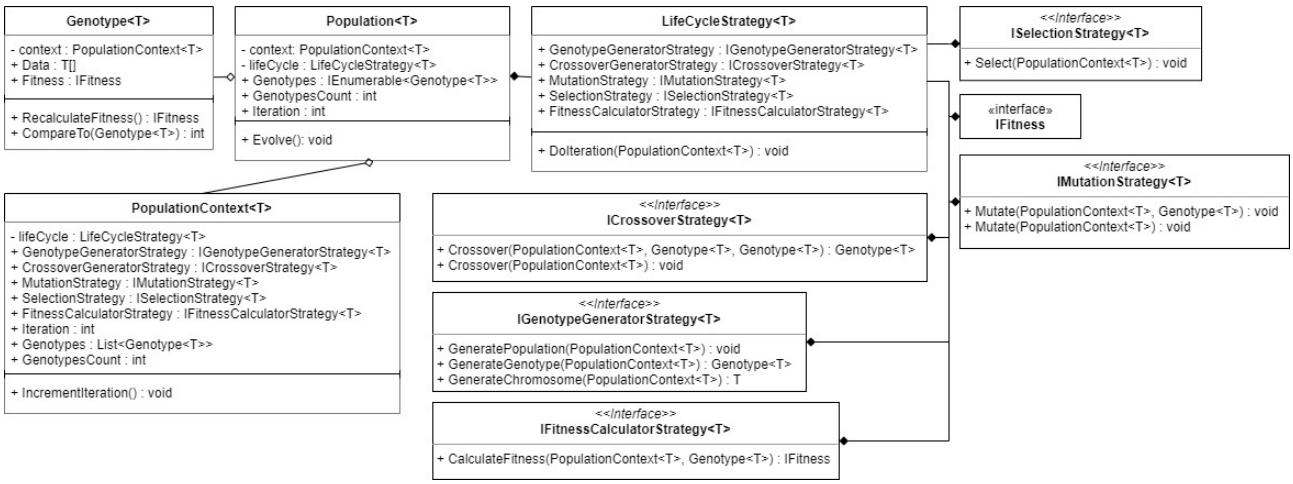


Рис. 4. Диаграмма классов приложения

6. Реализация алгоритмов укладки

Для решения поставленной задачи было разработано несколько реализаций алгоритма укладки.

6.1. Простейшая функция определения выживаемости

Простейшая реализация основана на жадном алгоритме. Создается матрица, с числом столбцов, равным ширине листа. Высоту матрицы представляет динамически расширяемый список. Каждая ячейка – это состояние поля (занято частью фигуры или свободно). Алгоритм тесно связан со способом укладки и представлен в виде блок-схемы на рис. 5. Сложность алгоритма квадратичная, так как будет рассмотрено все множество точек, для каждой будет проверено число ячеек, равное площади прямоугольника. Данный способ поиска решения неприменим для задач с большим числом фигур, ввиду своей неэффективности.

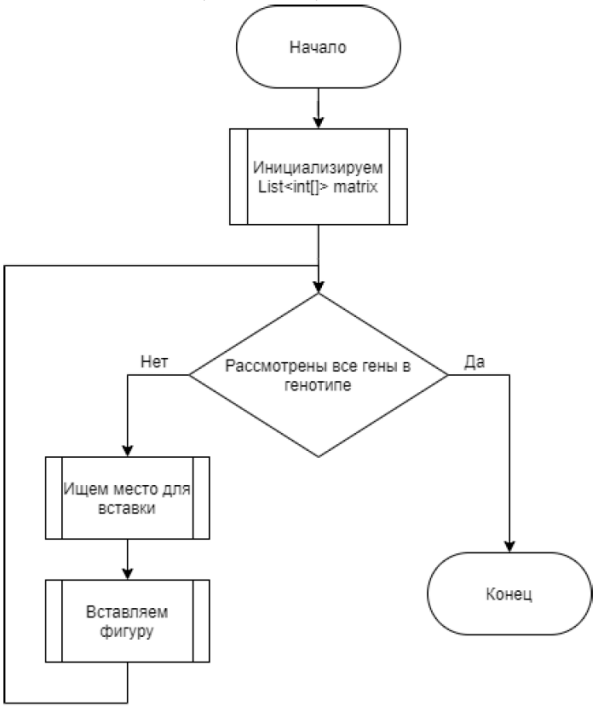


Рис. 5. Блок-схема алгоритма простейшего определения выживаемости

6.2. Быстрое определение выживаемости генотипа

Для ускорения размещения прямоугольников на плоскости необходимо отказаться от моделирования матрицы в соотношении 1:1. Поэтому был предложен алгоритм быстрого подсчета выживаемости, базирующийся на вычислении максимальных высот по всей ширине листа (рис. 6).

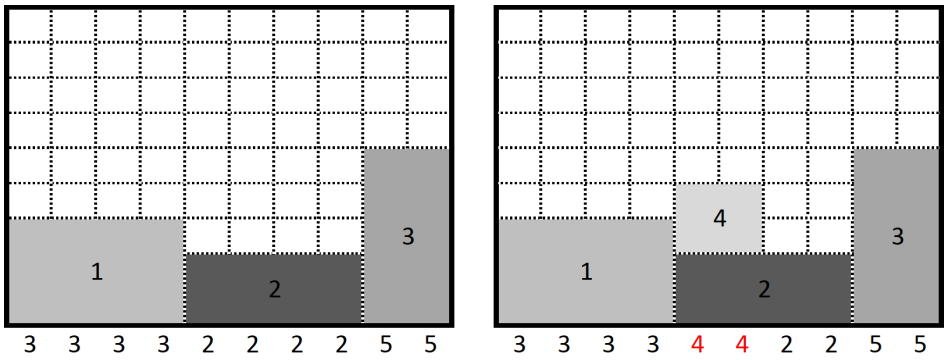


Рис. 6. Пример размещения прямоугольника в процессе быстрого определения выживаемости

1. Выбирается фигура с порядковым номером, находящимся в первой координате вектора-генотипа.

2. Для каждой точки ширины листа по площади фигуры вычисляется максимальная высота, на которую поднимется фигура, если будет установлена в эту точку.

3. Из чисел, полученных в пункте 2, выбирается минимальное.

4. Прямоугольник устанавливается в точку, в которой значение высоты было минимальным.

Для оптимизации поиска минимумов по ширине фигуры была использована структура данных дерево отрезков (рис. 7).

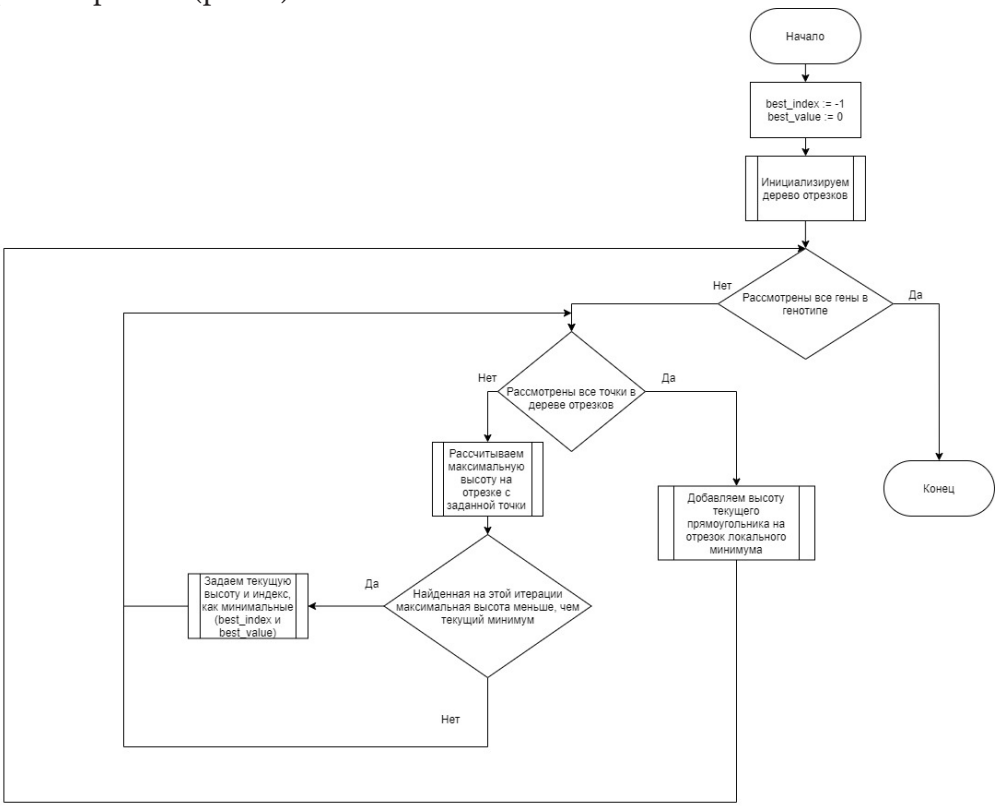


Рис. 7. Блок-схема алгоритма быстрого определения выживаемости

7. Результаты

Тестирование работоспособности созданного решения проводилось на специально подготовленных наборах прямоугольников. Они были получены из цельного прямоугольника заданного размера, путем деления его на части. Было получено 1235 прямоугольников из плоскости размером 20 × 10000. Размер популяции был фиксированным и состоял из 100 особей.

На рис. 8–10 представлены графики, отражающие зависимость значения фитнес-функции от числа итераций алгоритма (эпох). На графиках видно, что лучшее решение в популяции либо длительное время остается неизменным, либо меняется в лучшую сторону (так как не выполняется мутация для родительских генотипов). Поэтому траектория развития популяции сильно напоминает гиперболу в первой четверти.

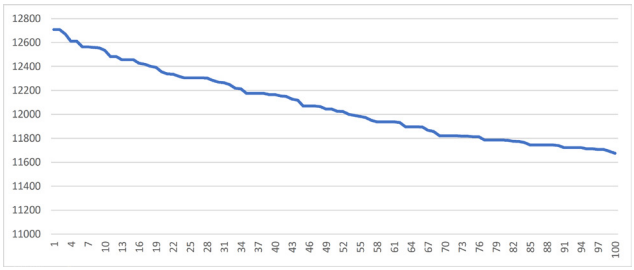


Рис. 8. Значение фитнес-функции для первых 100 популяций

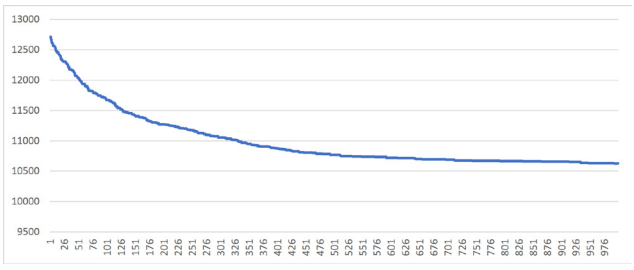


Рис. 9. Значение фитнес-функции для первых 1000 популяций

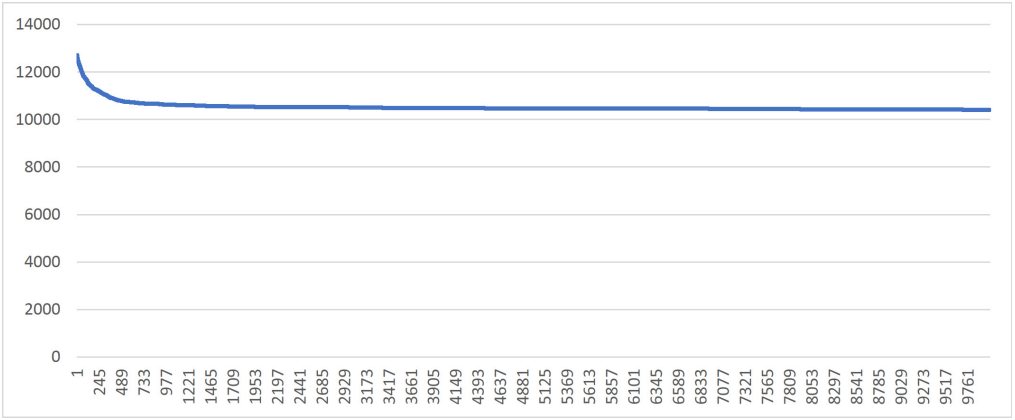


Рис. 10. Значение фитнес-функции для первых 10000 популяций

Заключение

В процессе исследования был реализован генетический алгоритм для решения задачи двумерной прямоугольной упаковки, основанный на идеях из [2], рассмотрен и спроектирован жизненный цикл популяции, разработан способ тестирования, позволяющий оценить оптимальность решения. Разработан быстрый алгоритм определения выживаемости, который в совокупности с предложенной фитнес-функцией позволяет получить сходимость решения к оптимальному за приемлемое время.

Литература

1. Martello S., Toth P. Knapsack Problems. – Wiley, Chichester, 1990.

2. Jakobs, S. On genetic algorithms for the packing of polygons / S. Jakobs // European Journal of Operational Research. – 1996. – Vol. 88. – P. 165–181.

Покатаев Николай Владимирович – студент 2-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: kolya21082001@gmail.com

Авсеева Ольга Владимировна (научный руководитель) – канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: olga-avseeva@mail.ru

УДК 616

ВИЗУАЛИЗАЦИЯ ЗРЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ КЛАССИФИКАЦИИ РАСПОСТРАНЕННЫХ БОЛЕЗНЕЙ ГРУДНОЙ КЛЕТКИ

В. В. Поляков

Воронежский государственный университет

Введение

Данная статья посвящена визуализации прохождения изображений через модель, созданную на основе алгоритмов глубокого обучения, которая была описана в [8].

Это представление поможет лучше понять механизм работы сверхточных нейронных сетей и описать причины, из-за которых архитектура имеет ту или иную оценку эффективности.

Метрики подобраны с учетом имеющихся данных.

1. Машинное обучение и Глубокое обучение

1.1. Машинное обучение

Машинное обучение – это научная область, изучающий методы построения алгоритмов, способных обучаться на основе имеющихся данных. Говорят, что компьютерная программа обучается на основе опыта E по отношению к некоторому классу задач T и некоторой мере эффективности P, если эффективность программы при решении задач из T, измеряемая с помощью P, повышается с опытом E (определение Машинного Обучения, Том Митчел, 1997).

Существует два основных класса алгоритмов машинного обучения:

- Обучение с учителем.
- Обучение без учителя.

В данном случае рассматривается задача классификации.

1.2. Глубокое обучение

Глубокое обучение – совокупность широкого семейства методов машинного обучения, основанных на имитации работы человеческого мозга в процессе обработки данных и создания паттернов, используемых для принятия решений.

Данный класс алгоритмов машинного обучения использует многослойную систему нелинейных фильтров, для извлечения признаков с преобразованиями, каждый последующий слой получает входные данные предыдущего слоя (рис. 1).

Слои моделей формируются в процессе обучения для выявления признаков на нескольких уровнях представлений, которые соответствуют различным уровням абстракции; при этом признаки организованы иерархически – признаки более высокого уровня являются производными от признаков более низкого уровня.

2. Базовая архитектура сверточных нейронных сетей

Сверточная нейронная сеть – специальная архитектура нейронных сетей, изначально нацеленная на эффективное распознавание изображений. Описание алгоритма свертки присутствует в предшествующей данной статье.

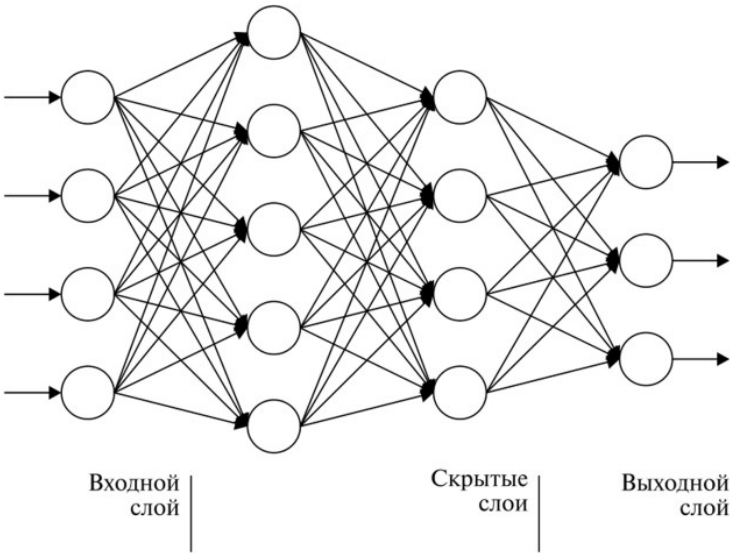


Рис. 1. Пример глубокой нейронной сети

Сверточный слой нейронной сети представляет из себя применение операции свертки к выходам предыдущего слоя, где веса ядер являются обучаемыми параметрами.

Пулинговый слой призван снижать размерность изображений, исходное изображение делится на блоки размером $w \times h$ и для каждого блока вычисляется некоторая функция.

3. Работа с данными

Этот набор данных представляет собой 112120 изображений, более чем 30000 различных пациентов. Данные могут принадлежать к 15 классам, при этом 14 из них – классы болезней. Одно изображение может одновременно иметь несколько меток классов. Основная проблема этого набора изображений – его несбалансированность. Так из 112120 изображений 51759 изображений не имеют болезней. Если отдельно выделить случаи, когда одно изображение имеет лишь одну метку класса болезни, можно увидеть, что наиболее объемный класс болезни (инфильтрация легких) имеет около 9500 изображений, а наименее объемный (грыжа) – менее тысячи.

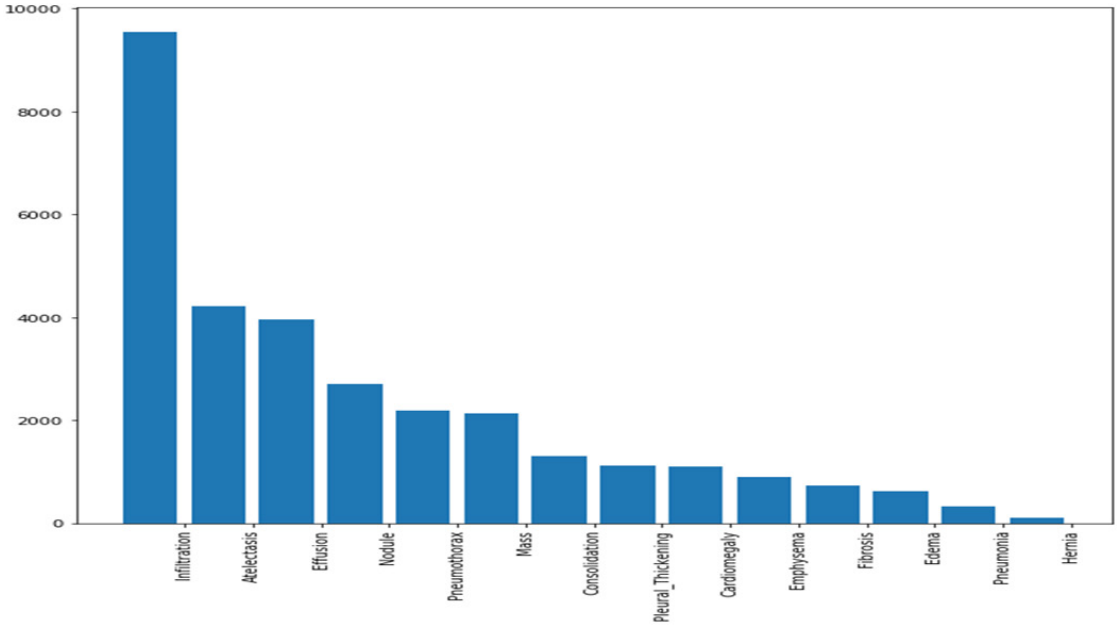


Рис. 2. Гистограмма частоты болезней

Также, как завил из работников медицинской сферы, далеко не все метки классов расставлены верно, что осложняет обучение нейронной сети.

4. Модели

4.1. Модель для мульти-классовой классификации

В случае второго подхода к созданию модели светочной нейронной сети, произведена попытка решить задачу мульти-классовой классификации. Для начала, уберем из набора данных изображения, помеченные как отсутствие болезней, будем полгать, что, если при предсказании не было найдено ни одного класса ни одного класса из предложенных – пациент здоров. При обучении, нейронная сеть показывает неплохие результаты метрики accuracy, как на тренировочном, так и на валидационном наборе (около 88 %). Однако, данная метрика не может показать правильность нейронной сети. Как и говорилось ранее, данный набор сильно не сбалансирован, это означает, что высокие значения accuracy можно получить, предсказывая мажоритарные классы. Данное заключение подтверждается выводом кривой ошибок.

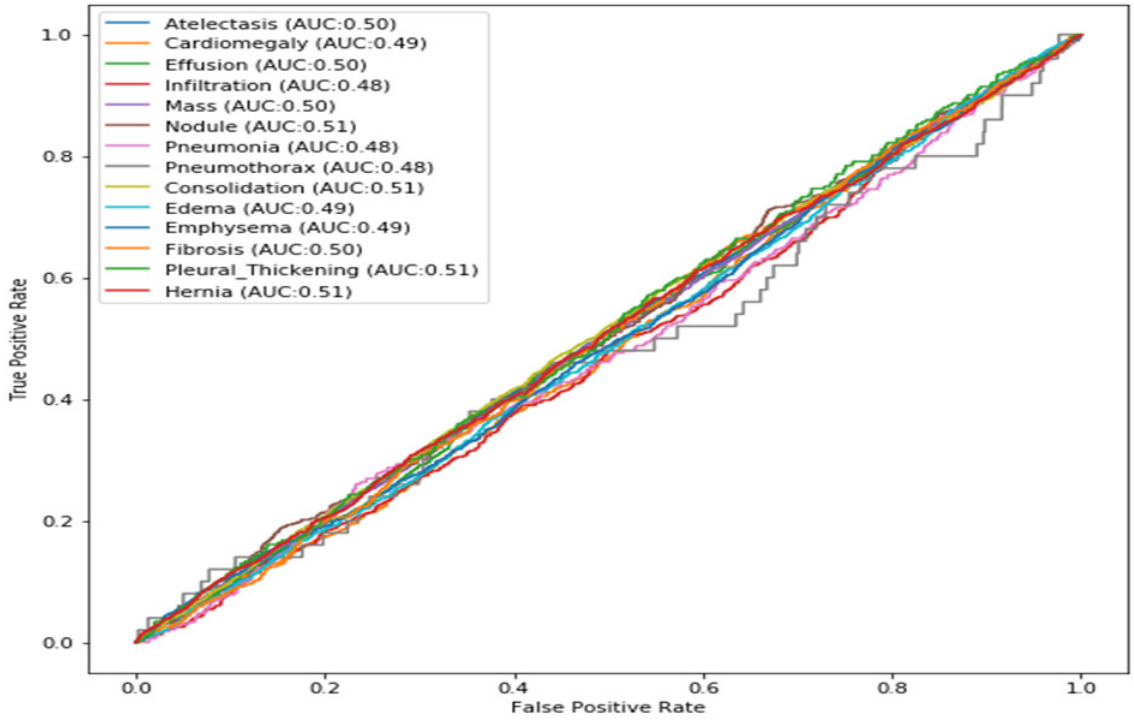


Рис. 3. Кривая ошибок для случая мульти-классовой классификации

4.2. Визуализация

Несмотря на то, что при использовании указанного набора данных нейронная сеть может предсказывать лишь недостоверные результаты, визуализация фильтров может помочь понять, на что обращается внимание при предсказании того или иного класса болезни. Также, в дальнейшем это может помочь изменить архитектуру модели для получения лучших результатов.

Для создания представления зрения модели загружается веса, полученные при обучении, затем удаляются полносвязные слои, ответственные. Вывод содержимого слоев, через которые проходит изображение и будет необходимой визуализацией.

В данном примере, было использовано верно классифицируемое изображение. На рисунке представлены лишь фильтры первых двух сверток и пулингов, так как вывод большего коли-

чества не несет никой полезной информации. С каждой парой слоев для обработки картинки, каналов становиться больше, когда как их размер уменьшается до минимального размера в 7 на 7 пикселей.

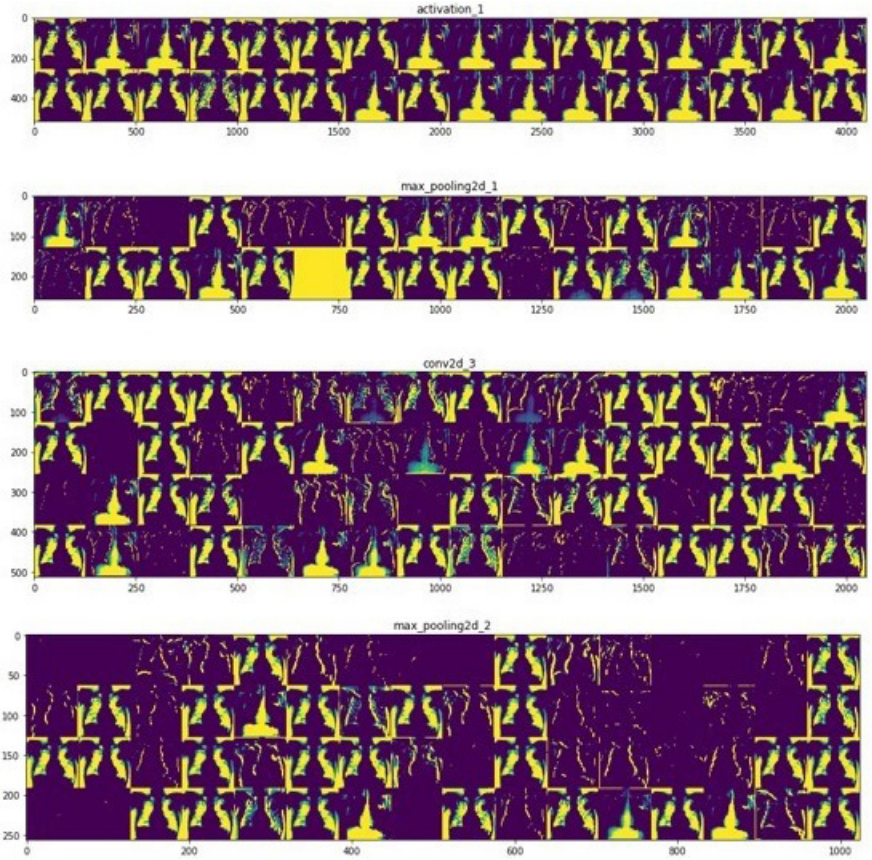


Рис. 4. Визуализация зрения модели при обработке изображения

Визуализация помогает понять, на какие части классифицируемого изображения обращается внимание при присваивании того или иного маркера. Как можно видеть, изначально акцент делается на выделении контрастности между цветами изображения, это делается для нахождения изначальных форм. В дальнейшем ищется более конкретная информация, связанная с отдельными частями изображения. Соответственно, чем глубже слой, тем более сложную сложные он несет. В данном случае очевидна причина плохих оценок метрик нейронной сети, еще на втором сверточном слое, после применения операции пулинга, теряется информация из более чем 25 фильтров, что может свидетельствовать о недобученнии.

Заключение

Несмотря на то, что свертка является мощным инструментом при создании нейронной сети для классификации изображений, ее моделям сложно обучаться на плохо подготовленных данных. Однако, несмотря на несовершенство данного набора данных, визуализация вывода фильтров помогает лучше разобраться в работе алгоритмов указанной архитектуры.

Литература

1. *Мюллер, А.* Введение в машинное обучение с помощью Python / А. Мюллер. С. Гвидо. – Москва, 2017. – 393 с.

2. Сайт контроля версий. – Режим доступа: https://github.com
3. Платформа для просмотра курсов онлайн. – Режим доступа: https://www.coursera.org
4. Документация фреймворка керас. – Режим доступа: https://keras.io
5. Документация фреймворка тензерфлоу. Режим доступа: https://www.tensorflow.org
6. Платформа для даса саентистов кэггл. Режим доступа: https://www.kaggle.com
7. Сайт для публикации статей о дата саенс. Режим доступа: https://towardsdatascience.com
8. *Поляков В.* Анализ медицинских изображений с использованием сверточных нейронных сетей для классификации распространённых болезней грудной клетки.

Поляков Вадим Витальевич – студент 4-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: vadpolyoab@gmail.com

Светлана Юрьевна Болотова (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: bolotova.svetlana@gmail.com

УДК 004.3

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА КАНАЛОВ СВЯЗИ МНОГОФУНКЦИОНАЛЬНОГО БЛОКА УПРАВЛЕНИЯ И КОНТРОЛЯ ОСНОВНЫХ СИСТЕМ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА СЕМЕЙСТВА AVR СПОСОБНОГО ФУНКЦИОНИРОВАТЬ В СОВРЕМЕННОМ АВТОМОБИЛЕ

В. А. Пометов

Воронежский государственный университет

Введение

Можно считать, что микроконтроллер – это компьютер, разместившийся в одной микросхеме. Отсюда и его основные привлекательные качества: малые габариты; высокие производительность, надежность и способность быть адаптированным для выполнения самых различных задач. В системе современного автомобиля находится большое количество электронных блоков управления, основным вычислительным элементом которых является микроконтроллер.

При проектировании блока управления и контроля основных систем автомобиля по средствам микроконтроллера семейства AVR и последующей разработки соответствующего программного обеспечения проводится анализ уже имеющихся в системе современного автомобиля блоков управления и их полную функциональность. Такой анализ позволяет разобрать различные процессы и прогнозировать их развитие. В настоящее время данный подход широко используется при проектировании и разработке программного обеспечения дополнительно внедряемых в систему современного автомобиля блоков контроля и управления, с целью расширения и улучшения возможную функциональность.[1]

1. Постановка задачи

На данном этапе исследования требуется изучить имеющиеся на борту современного автомобиля блоки электронного управления и их программное обеспечение, позволяющее функционировать всем устройствам и системам. В качестве основных задач можно выделить три главных.

- Сбор необходимых данных, о количестве и функциональности электронных блоков управления имеющихся на борту автомобиля.
- Исследование их функционала, способах связи между собой и с основными системами автомобиля.
- Проектирование стороннего блока контроля и управления на базе микроконтроллера семейства AVR, с подходящими к имеющимся блокам и системам в автомобиле каналами связи, управления и обмена данными.

Исследование, которое необходимо сделать в данной работе, представляет собой общее изучение протоколов связи и контроля электронных блоков управления систем автомобиля, разбор функциональности имеющихся

2. Анализ

Перед началом разработки собственного блока управления и контроля электронных периферийных устройств в системе современного автомобиля обязательно необходимо опреде-

лить количество, устройство, программное обеспечение и протоколы связи уже имеющихся на борту автомобиля электронных блоков управления (рис. 1). Самый простой и доступный способ – это проанализировать принципиальную электронно-техническую схему конкретного автомобиля и выделить все основные электронные блоки управления системами и периферийными устройствами внутри автомобиля, а также определить вычислительные мощности каждого из них, определив управляющий системой микроконтроллер внутри электронного блока управления.

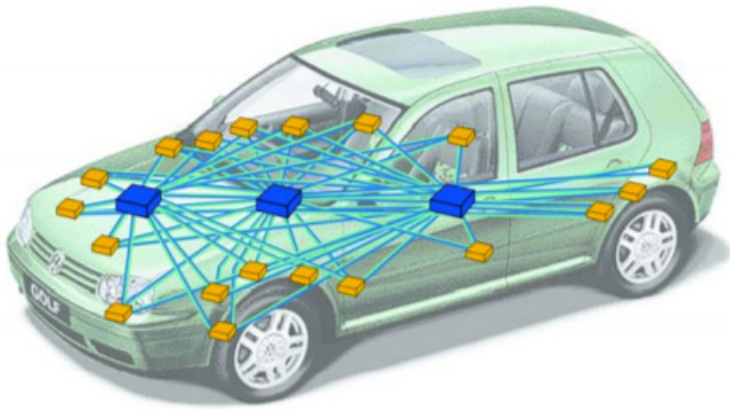


Рис. 1. Схематичное изображения ЭБУ внутри автомобиля и каналы их связи

3. Классификация средств обмена данными

В первую очередь, чтоб понять, как происходит обмен данными в бортовой системе автомобиля между электронными блоками управления необходимо разобрать основные способы связи. Наиболее распространёнными являются:

1) Проводные интерфейсы связи внутри одной подсети.

• SPI – способ связи с использованием приёмопередающих устройств, предназначенный для обеспечения простого и надежного сопряжения микроконтроллеров с другими устройствами.

• UART – интерфейс, предназначенный для организации связи с другими устройствами. Преобразует передаваемые данные в последовательный вид так, чтобы было возможно передать их по цифровой линии другому аналогичному устройству

2) Проводные интерфейсы для связи отдельных сетей между собой.

• RS-485 – обеспечивает обмен данными между несколькими устройствами по одной двухпроводной линии связи.

• CAN – это последовательный протокол связи с эффективной поддержкой распределения контроля в реальном времени и очень высоким уровнем безопасности. Основное назначение: организация передачи информации в сложных условиях, таких как среды с высоким уровнем различного рода помех.

3) Беспроводные интерфейсы передачи данных.

• Wi-Fi – протокол и стандарт на оборудование для широкополосной радиосвязи, предназначенной для организации локальных беспроводных сетей.

• Bluetooth – представляет собой беспроводной интерфейс, характеризующийся малым радиусом действия, а также не требует больших ресурсов в плане потребления энергии.

• Модуль Bluetooth RN-42 предназначен для замены кабельного соединения между двумя устройствами, связанными по последовательному асинхронному интерфейсу (UART). Он позволяет организовать передачу данных по беспроводной технологии Bluetooth на расстояние до 10–20 метров.

Изучив принципиальные схемы подключения электронных блоков управления внутри современных автомобилей, можно выявить основную тенденцию среди способов связи как внутри одного основного блока, так и между отдельными блоками контроля и управления системами автомобиля: внутри одного блока или подсети используется интерфейс UART, между различными блоками и для связи с периферийными устройствами используется интерфейс передачи данных CAN (рис. 2), а для беспроводной связи между контроллерами и устройствами, а также внешними устройствами используется Bluetooth [2].

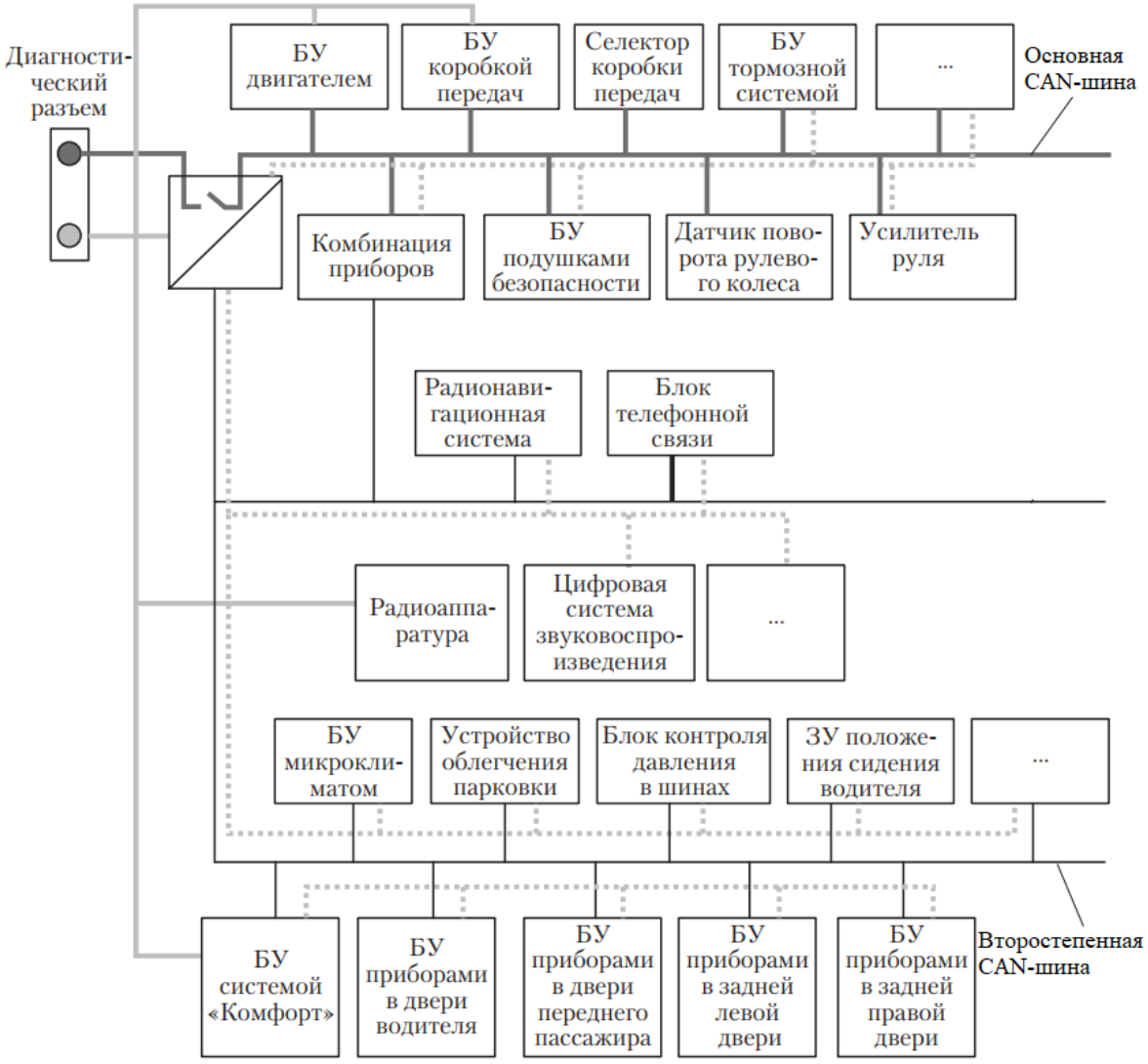


Рис. 2. CAN-шина соединения ЭБУ в автомобиле

4. Проектирование принципов связи стороннего электронного блока управления

Проанализировав вариации и способы связи между имеющимися на борту автомобиля электронными блоками управления можно выделить аналогичные способы связи и для проектируемого дополнительного блока.

Для связи с внешними сетями на борту автомобиля лучше всего использовать CAN-интерфейс, так как все иные электронные блоки управления используют его. Для абстрагирования от среды передачи спецификация CAN избегает описывать биты данных как «0» и «1». Вместо этого применяются термины «рецессивный» бит и «доминантный» бит, при этом подразумевается, что при передаче одним узлом сети рецессивного бита, а другим доминантного

принят будет доминантный бит. Например, при реализации физического уровня на радиоканале отсутствие сигнала означает рецессивный бит, а наличие – доминантный. Таким образом организация передачи данных через CAN-интерфейс позволяет при необходимости с минимальными настройками внедрить разрабатываемый блок управления в любой автомобиль, использующий для обмена информации между подсетями CAN-интерфейс, без значительной перенастройки или глобального изменения программного обеспечения [3].

Для реализации передачи данных внутри программируемого блока по аналогии с уже установленными на автомобиль электронными блоками управления используется UART интерфейс. Он преобразует передаваемые данные в последовательный вид так, чтобы было возможно передать их по одной физической цифровой линии другому аналогичному устройству. Метод преобразования хорошо стандартизован и широко применяется в компьютерной технике. Многие реализации UART имеют возможность автоматически контролировать целостность данных методом контроля битовой чётности. Помимо информационных битов, UART автоматически вставляет в поток синхронизирующие метки, так называемые стартовый и стоповый биты.[4] UART-протокол прост и надежен. Реализация внутри электронного блока передачи информации таким способом отвечает всем требованиям, а также позволяет модернизировать и добавлять дополнительные модули без преобразования и переработки всей архитектуры устройства и без значительных изменений внутри программного обеспечения.[5]

Для беспроводного подключения других периферийных устройств предпочтительнее всего Bluetooth, за счет общей популярности этого способа связи. Помимо подключения устройств для обмена информации по беспроводной связи, можно также подключить устройство управления, например, сотовый телефон. Для управления разрабатываемым блоком необходимо специальное приложение на управляющем устройстве, позволяющее отправлять команды, воспринимаемые блоком управления за счет специальных настроек в программном обеспечении.

Заключение

В рамках данной работы было проанализировано и подробно разобрано устройство электронных блоков управления, находящихся на борту современного автомобиля. Это необходимо для проектирования многофункционального блока управления и контроля основных систем и периферийных устройств с помощью микроконтроллера семейства AVR, способного функционировать в связке с уже имеющимися ЭБУ автомобиля, не нарушая принципов их работы. Результатом данной работы можно считать полученный набор принципов и правил соединения каналов связи внутри управляемого блока, а также подключения иных периферийных устройств.

Литература

1. *Алексеев В. Е.* Микроконтроллеры. Структуры данных: Учебник / В. Е.Алексеев, В. А. Таланов. – Нижний Новгород : Изд-во ННГУ, 2005 – 307 с.

2. *Оре О.* Микроконтроллеры семейства AVR / О. Оре. – М. : Едиториал УРСС, Ленанд, 2015. – 208 с.

3. *Koenig Sven, Maxim Likhachev, Yaxin Liu*. Coder's Guide to IoT Development. – 2004. – 99. – 112 с.

4. *Лафоре Р.* Структуры данных и алгоритмы AVR-микроконтроллера. Классика Computer Science. 2-е издание. – СПб. : Питер, 2011. – 704 с.

5. *Филипс Б.* Справочник по электрооборудованию. Устройства защиты и управления. Электрические устройства / Б. Филипс, Б. Харди. – Санкт-Питербург : Изд-во Питер, 2014. – 712 с.

Пометов Валерий Александрович – студент 2-го курса магистратуры кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: pometow@gmail.com

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: Bolotova.svetlana@gmail.com

УДК 004.8

ПРОГНОЗИРОВАНИЕ ЗАГРУЖЕННОСТИ ПАРКОВОК НА ОСНОВЕ РЕКУРРЕНТНЫХ СЕТЕЙ

А. А. Поминова

Воронежский государственный университет

Введение

Одна из наиболее востребованных сфер получения и использования информации – сфера дорожного движения. На сегодняшний день существуют программные продукты, позволяющие отслеживать различные данные, связанные с состоянием дорожного движения: степень пробок на дорогах, отслеживание движения общественного транспорта и в частности просмотр информации о парковках и доступных на них местах. Однако не все из них предоставляют данные необходимые пользователю. Большинство таких приложений нацелены на отображение адресов и местоположений коммерческих парковок. Существуют так же некоторые сервисы и приложения, которые предоставляют информацию и о текущем состоянии парковок города. В настоящее время в городах очень актуальна проблема поиска свободного парковочного места. И пользователь, который получает информацию о текущем количестве свободных мест на интересующей его парковке, не может быть уверен, что ситуация останется прежней, когда он доберется на место.

Решением такой проблемы может стать сервис, который предоставляет прогноз о количестве свободных мест на выбранной пользователем парковке. В статье рассматривается вариант, при котором информация о загруженности парковки представлена в виде одномерного временного ряда. А для прогнозирования следующего значения такого временного ряда используется рекуррентная нейронная сеть.

1. Теория

1.1. Временные ряды

Временной ряд – собранный в разные моменты времени статистический материал о значении каких-либо параметров исследуемого процесса. Каждая единица статистического материала называется измерением. При анализе данных временного ряда учитывается взаимосвязь измерений со временем, поэтому важно соблюдать правильную последовательность данных во времени. По количеству измерений временные ряды делятся на одномерные и многомерные. В контексте рассматриваемой задачи в качестве основного измерения временного ряда выступает количество свободных мест на парковке. Дорожный трафик и загруженность парковок в городе зависят от сочетания разных показателей, например, день недели, время суток, погодные условия, район города, где находится парковочное место. Таким образом, для более точного прогнозирования можно производить вычисления над многомерным рядом, состоящим из вышеперечисленных измерений. В данной статье приведен пример вычислений с использованием одномерного временного ряда.

1.2. Рекуррентные нейронные сети

Рекуррентная нейронная сеть (РНС) – это тип НС, который хорошо подходит для решения задач, связанных с временными рядами. РНС шаг за шагом обрабатывает временную последо-

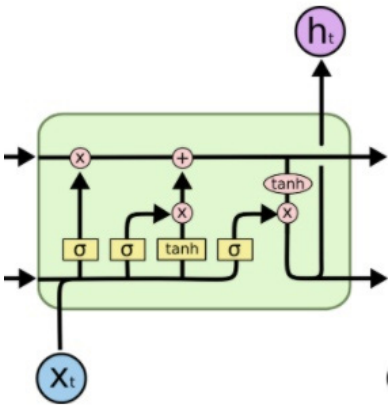


Рис. 1. Пример LSTM ячейки

вательность данных, перебирая ее элементы и сохраняя внутреннее состояние, полученное при обработке предыдущих элементов. LSTM (Long short-term memory) – разновидность РНС, особенность архитектуры которой заключается в том, что в слоях этой сети находятся не нейроны, а так называемые «ячейки памяти».

Ячейка LSTM, представленная на рис. 1, состоит из нескольких вентилей, которые определяют, осталась ли старая информация актуальной и является ли новая информация значимой для текущей задачи прогнозирования. Такая память называется состоянием ячейки и позволяет сохранять информацию, полученную намного раньше в последовательности. Эта особенность делает LSTM очень мощной для анализа последовательностей, таких как временные ряды.

2. Решение

2.1. Модель данных

Временной ряд представлен в следующем виде: $\{x_i\}_{i=1}^{N}$, $x(t)$ – наблюдаемая динамическая переменная с некоторым постоянным шагом τ по времени, $t_i = t_0 + (i-1)\tau$: $x_i = x(t_i)$, $i = 1,...,N$, где x_i – количество свободных парковочных мест в момент времени t_i, с шагом $\tau = 0.5$ часа, начальный момент времени t_0 = 01.01.2019 0:00. Входной набор данных представляет собой временной ряд, содержащий количество свободных парковочных мест в течение года с замерами каждые полчаса.

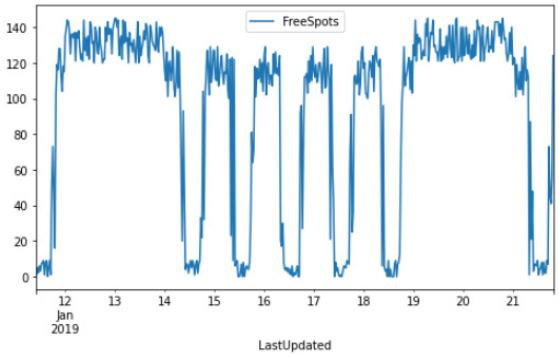


Рис. 2. Визуализация входных данных

Входные данные записаны в файле с расширением .csv, и для одномерного временного ряда содержат колонки: LastUpdated – временная отметка, FreeSpots – количество свободных мест. Данные для наглядности представлены в виде графика, который можно видеть на рис. 2. По нему можно заметить, что количество свободных мест возрастает в выходные и вечером в рабочие дни, и сокращается в течение рабочего дня.

2.2. Программное решение

Для решения поставленной задачи использовалась библиотека Keras. Конфигурация LSTM сети представлена на рис. 3.

```
lstm_model = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(8, input_shape=x_train_uni.shape[-2:]),
    tf.keras.layers.Dense(1)
])

lstm_model.compile(optimizer='adam', loss='mae')
```

Рис. 3. Конфигурация LSTM сети

1) LSTM – слой с восемью LSTM ячейками;

2) Dense – выходной слой с одним выходным значением;

3) adam – метод стохастического градиентного спуска, основанный на адаптивной оценке первого и второго порядков;

4) mae – средняя абсолютная ошибка регрессионной потери. Метрика для мониторинга процесса обучения.

После обучения модели ее можно использовать для прогнозирования значений. Один из таких прогнозов представлен на рис. 4.

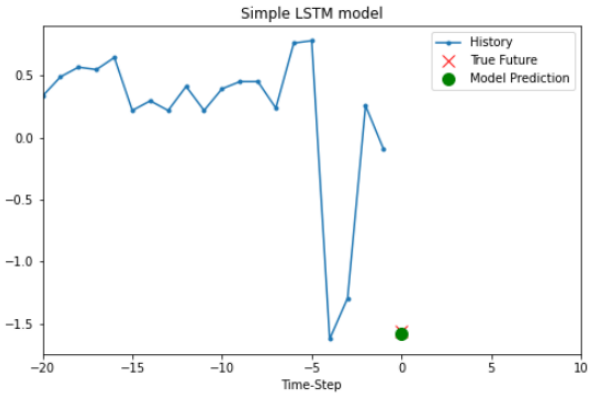


Рис. 4. Пробный прогноз модели

Заключение

В данной статье был представлен вариант решения задачи о прогнозировании загруженности парковки. Данные были представлены в виде одномерного временного ряда, где измерением является количество свободных парковочных мест для заданной временной отметки. Это позволило использовать рекуррентную нейронную сеть для решения поставленной задачи. В результате был получен прогноз о количестве свободных парковочных мест на следующий час.

Литература

1. *Орельен, Ж.* Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. Концепции, инструменты и техники для создания интеллектуальных систем / Ж. Орельен. – 2-е изд. – Москва : Вильямс, 2018. – 688 с.

2. *Шолле, Ф.* Глубокое обучение на Python / Ф. Шолле. – Санкт-Петербург : Питер, 2018. – 400 с.

3. *Будума, Н.* Основы глубокого обучения. Создание алгоритмов для искусственного интеллекта следующего поколения / Н. Будума. – Москва : МИФ, 2020. – 304 с.

4. Прогнозирование временных рядов. – Режим доступа: https://www.tensorflow.org. – (Дата обращения: 14.03.2021).

5. LSTM layer. – Режим доступа: https://keras.io. – (Дата обращения: 05.04.2021).

Поминова Алена Андреевна – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.

E-mail: alena.pominova@gmail.com

Горбенко Олег Данилович (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: oleg_dan@mail.ru

УДК 004.021

ИССЛЕДОВАНИЕ ЗАДАЧИ ПОИСКА ТОЧЕК РАЗМЕЩЕНИЯ АВТОМОБИЛЕЙ КАРШЕРИНГА И КОШЕРИНГА

Е. А. Попова, Д. В. Борисенков

Воронежский государственный университет

Введение

В обществе все больше и больше набирает скорость движение в сторону экономики совместного потребления или шеринг-экономики, аренда имущества начинает преобладать над собственностью, переиспользование и разделение ресурсов становится все более популярным. Появились и развиваются сервисы краткосрочной аренды различного транспорта: велошеринг, каршеринг и другие.

Расставить единицы транспорта по городу, чтобы покрыть все возможные районы с минимальными затратами – достаточно сложная задача. В данной работе рассматривается задача поиска точек размещения автомобилей каршеринга и кошеринга.

Задача поиска точек размещения автомобилей

Прежде, чем рассматривать основные алгоритмы, которые используются для достижения поставленной задачи, необходимо определить входные данные, их формат, методы их обработки, а также ввести необходимые определения.

В качестве анализируемых данных используются открытые данные общественных карт, такие как маршруты и остановки общественного транспорта, а также общедоступные GPS-треки – последовательности координат, объединенные в списки, представляющие собой перемещение человека в некоторый промежуток времени. Пример отображения GPS-треков на карте представлен на рис. 1.

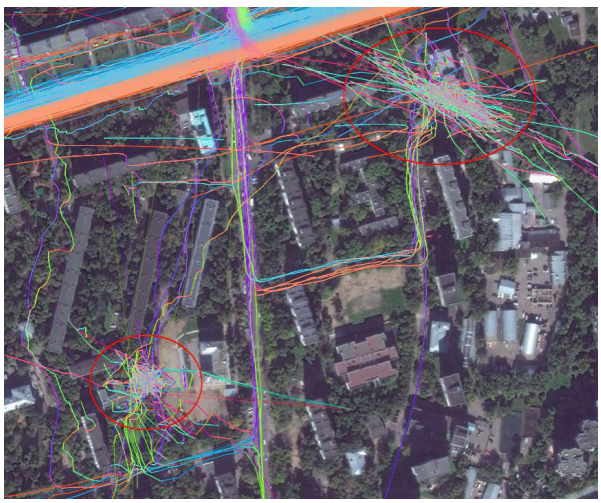


Рис. 1. Отображение GPS-треков на карте

Размещение точек производится в ограничивающем прямоугольнике (bounding box) – заданной прямоугольной области, границы которой задаются в следующем порядке: крайняя южная широта, крайняя западная долгота, крайняя северная широта, крайняя восточная долгота. Например, ограничивающему прямоугольнику [55.62, 37.32, 37.87, 55.85] соответствует область, изображенная на рис. 2, а на рис. 3 – прямоугольнику [46.20, 47.74, 48.29, 46.49].

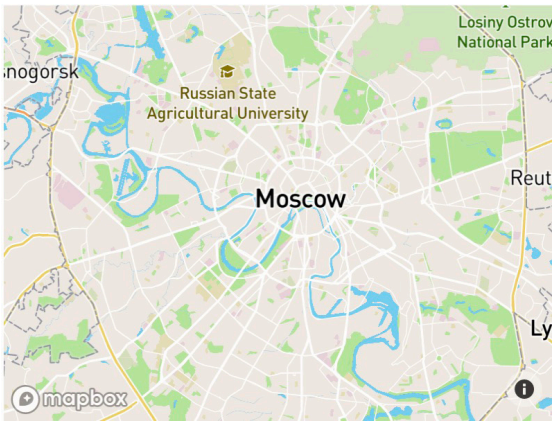


Рис. 2. Область, соответствующая огр. прямоугольнику [55.62, 37.32, 37.87, 55.85]

Рис. 3. Область, соответствующая огр. прямоугольнику [46.20, 47.74, 48.29, 46.49]

В заданном ограничивающем прямоугольнике производится поиск пользовательских GPS-треков, маршрутов и остановок общественного транспорта. С целью уменьшения объема входных данных, среди треков выделяются близкие друг к другу, т. е. координаты которых находятся между собой на минимальном заданном расстоянии, которое определяется заранее пользователем системы.

Далее определяются районы для размещения точек каршеринга. Такими районами будут считаться области, содержащие большое количество начальных и конечных точек GPS-треков в объединении с малым количеством маршрутов общественного транспорта.

Будем считать, что в заданной области малое количество маршрутов, если расстояние от точки до ближайшей остановки общественного транспорта больше минимального заданного значения.

Точки размещения каршеринга должны находиться на некотором оптимальном расстоянии от районов или даже внутри их области. Их расстановка может быть решена с помощью задачи о размещении объектов или задачи k-центра [1], которая является разделом вычислительной геометрии и исследования операций, исследующая оптимальное расположение объектов с целью минимизации цены перевозок с учетом ограничений.

В базовой формулировке задача о размещении объектов состоит из потенциальных точек размещения L, где объекты могут быть открыты и точек D, которые должны быть обслужены. Цель – выбрать подмножество F точек размещения объектов с целью минимизации суммы расстояний от каждой точки обслуживания до ближайшего обслуживающего объекта.

В терминах этой задачи объектами размещения являются станции автомобилей каршеринга, точки, которые должны быть обслужены – точки районов для размещения, а требования к расстоянию между станциями до районов будут выступать ограничениями.

Для решения используется вариант минимаксного размещения. Существуют различные алгоритмы ее решения: метод плит, аппроксимация $1+\varepsilon$, выделения отдаленных точек.

Метод плит основывается на методе «разделяй и властвуй» и начинается с разделения множества точек на 2 подмножества A и B, как представлено на рис. 4 [2]. Истинное решение изображено на рис. 5, центры – точки в круге радиуса r.

На следующем шаге вводятся линии (рис. 6), которые в терминах задачи образуют плиту. Они делят k-центры решения на 3 подмножества.

Если удалить те решения, которые находятся непосредственно внутри границ плиты, то видно, что решения из множества X не покрывают точки из множества B, а решения из Y – из множества A. Из этого выходит независимое свойство плиты, гарантирующее, что решения независимы, т. е. задача может быть разделена подзадачи. В итоге множество будет разбито таким образом, как изображено на рис. 7 [2].

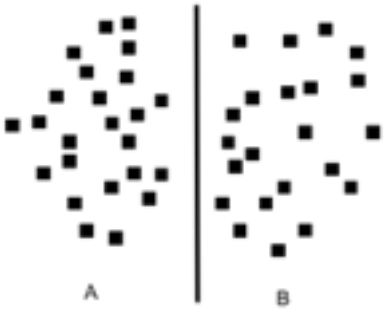


Рис. 4. Начало метода плит

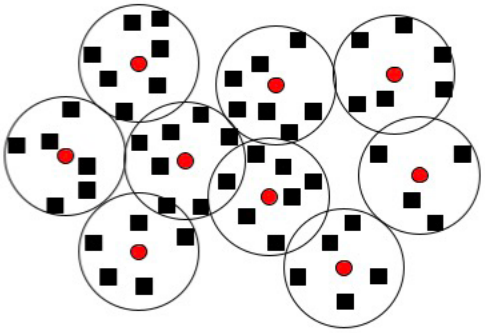


Рис. 5. Истинное решение

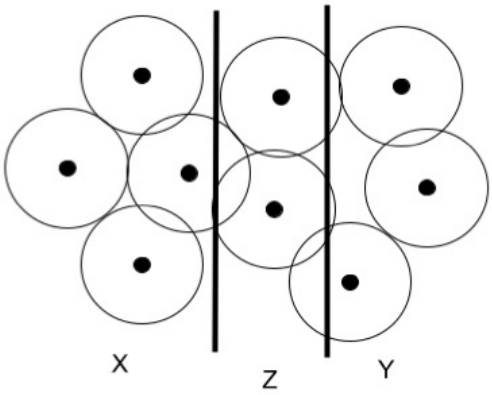


Рис. 6. Разделение на подмножества

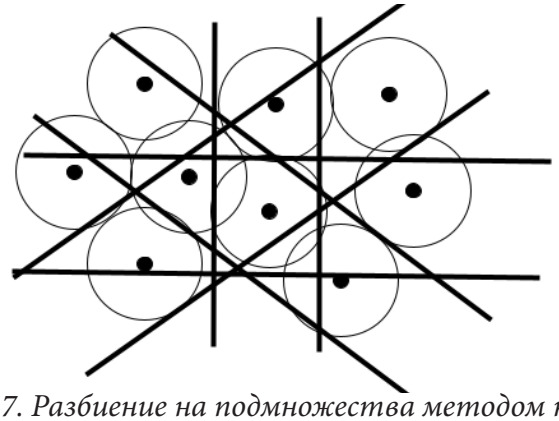


Рис. 7. Разбиение на подмножества методом плит

В случае метода аппроксимации $1+\varepsilon$ задается пустое результирующее множество M, а ε – желаемая точность.

Основной цикл алгоритма, проверяет, не покрыто ли уже множество точек $1+\varepsilon$ расширение B, т. е. не достигнута ли желаемая точность, в этом случае алгоритм завершается. Иначе должны быть точки вне $1+\varepsilon$ расширения B. В этом случае самая дальняя точка B рекурсивно помещается в каждую из M_i, где M_i – подмножество M, и вычисление k-центров производится рекурсивно [3].

Метод выделения отдаленных точек (farthest-first traversal) заключается в выборе произвольной стартовой точки, а каждая следующая – наиболее отдаленная от предыдущей (рис. 8). С помощью такого обхода находится k-центр [4]. Данное решение не будет являться идеально точным, но достаточно оптимальным [5].

Было проведено тестирование этих алгоритмов, реализованных на языке Python. В окончательной версии приложения по результатам дополнительного тестирования будет оставлен только наиболее эффективный алгоритм.

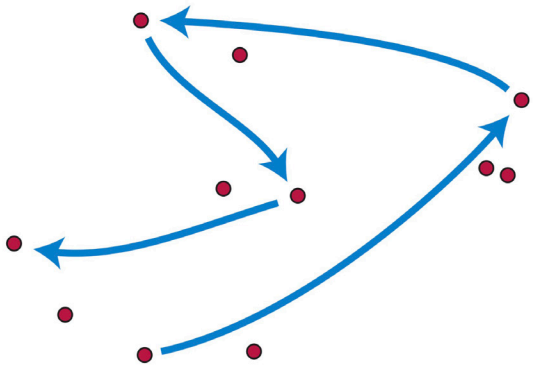


Рис. 8. Метод выделения отдаленных точек

Результатом работы алгоритма является набор точек, в которых можно разместить автомобили каршеринга, как это представлено на рис. 9.

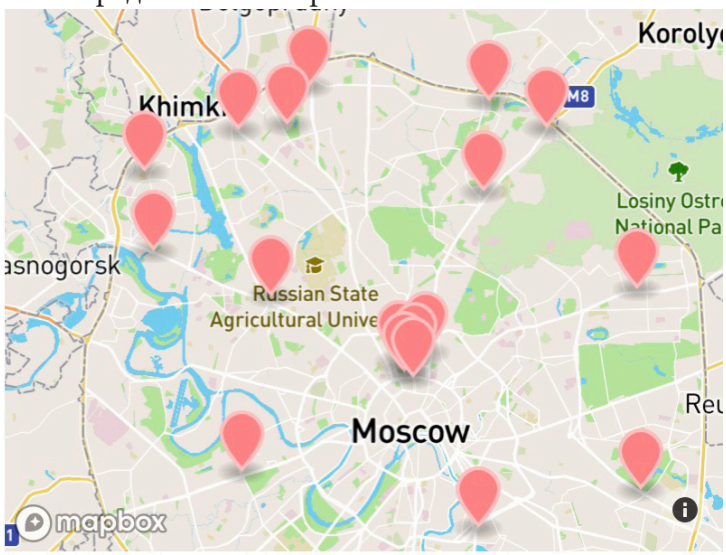


Рис. 9. Результат работы алгоритма

Заключение

В результате работы было проведено исследование алгоритмов размещения точек каршеринга и кошеринга, а также реализовано приложение, решающее данную задачу и имеющее наглядное визуальное отображение результата работы программы.

Данное приложение может быть использовано как компаниями, предоставляющими услуги каршеринга, так и обычными пользователями, желающими предоставить свой автомобиль для краткосрочной аренды автомобилей.

Литература

1. *Препарата, Ф.* Вычислительная геометрия: Введение / Ф. Препарата, М. Шеймос. Пер. с англ. – М. : Мир, 1989. – 478 с.

2. *Hwang, R. Z.* The slab dividing approach to solve the Euclidean p-center problem / R. Z. Hwang, R. C. T. Lee, R. C. Chang // Algorithmica. – 1993. – Vol. 9(1). – P. 1–22.

3. Kumar, P. Almost optimal solutions to k-clustering problems / Kumar, Pankaj; Kumar Piyush // International Journal of Computational Geometry & Applications. – 2010. – Vol. 20(4). – P. 431–447.

4. Gonzalez, T. Clustering to minimize the maximum intercluster distance / T. Gonzalez // Theoretical Computer Science – 1985. – Vol. 38. – P. 293–306.

5. *Dasgupta, S.* Clustering in metric spaces / S. Dasgupta, M. Paturi // CSE 291: Geometric algorithms – 2013. – P. 1–4.

Попова Елена Александровна – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: interna.carotis@gmail.com

Борисенков Дмитрий Васильевич (научный руководитель) – канд. техн. наук, доц., доц. кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: xuser@relex.ru

УДК 004.932.2

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА SMARTWALL

М. А. Принев

Воронежский государственный университет

Введение

Основной целью представляемого проекта является создание на основе разработанного автором метода сегментации зашумленных изображений с плавающим порогом бинаризации программного обеспечения SmartWall, позволяющего использовать компьютерное зрение для применения любой поверхности в качестве интерактивного элемента без использования сенсорных технологий.

Использование ПО SmartWall в образовательных учреждениях, а также в центрах реабилитации инвалидов позволит повысить уровень эффективности и доступности образования с использованием интерактивных IT технологий.

1. Описание базовой технологии

При решении задач, связанных с использованием компьютерного зрения, немаловажным является вопрос выбора метода сегментации изображения. Сегментация является одним из основных элементов работы автоматизированной системы технического зрения, т.к. именно на этой стадии обработки объекты выделяются из сцены для дальнейшего распознавания и анализа [1].

Научная новизна предлагаемых в инновационном проекте решений заключается в создании на основе разработанного автором метода сегментации зашумленных изображений с плавающим порогом бинаризации программного обеспечения SmartWall, позволяющего использовать компьютерное зрение для применения любой поверхности в качестве интерактивного элемента без использования сенсорных технологий. Сущность метода заключается в том, что порог бинаризации не определяется, как значение яркости пикселя, удовлетворяющее каким-либо условиям, а весь процесс бинаризации осуществляется посредством постоянного нарастания значения яркости пикселя, до тех пор, пока программа не сможет сегментировать заданное изображение, после этого изменение порога прекращается и фиксируется требуемое для данного изображения значение порога бинаризации, позволяющее сегментировать на нем нужную область.

Интерактивные поверхности при использовании ПО SmartWall могут быть двух типов:

- стационарные (программа запоминает координаты каждой интерактивной области и ее характеристики), достоинствами которых является более высокое быстродействие и надежность;
- мобильные (программа запоминает характеристики так называемого «якоря» и местоположение остальных интерактивных областей определяются алгоритмически), достоинствами которых является более разнообразные варианты использования и независимость от случайного смещения поверхности.

В качестве интерактивного элемента могут быть использованы поверхности любого цвета и текстуры при различных уровнях освещенности. Схемы работы ПО SmartWall для различных поверхностей представлены на рис. 1 и рис. 2.

Рис. 1. Схема работы ПО SmartWall для стационарных интерактивных поверхностей

Рис. 2. Схема работы ПО SmartWall для мобильных неэлектронных гаджетов

2. Требования к программному и аппаратному обеспечению

Основные конструктивные и технико-эксплуатационные показатели: для использования ПО SmartWall необходим компьютер ОС Windows или ОС Linux, одноплатный компьютер Raspberry Pi, не ниже модели 3B, веб-камера (разрешение: 1280x720; максимальная частота кадров: 30 Гц).

Технические параметры программных продуктов: быстродействие 2–5 с; объем памяти процессора, занимаемый приложением 60–75 МБ.

3. Реализация программно-алгоритмических задач

3.1. Оптимизация кода программы

В рамках поставленной задачи были проведены работы по оптимизации алгоритмического решения для программного обеспечения на основе метода *сегментации зашумленных изображений с плавающим порогом бинаризации*, позволяющего использовать компьютерное зрение для применения любой поверхности в качестве интерактивного элемента без использования сенсорных технологий. Принцип работы метода представлен на рис. 3, 4.

Оптимизация кода программы SmartWall была осуществлена за счет разработки и применения на этапе бинаризации изображения многотактного алгоритма, позволяющего осуществлять ячеистую обработку изображения при каждом проходе, что значительно увеличило быстродействие программы. В результате оптимизации были получены показатели, превосходящие до 5 раз аналогичные характеристики первоначального варианта кода программы. Скриншоты программы по сегментации зрачка глаза представлены на рис. 5.

Было найдено алгоритмическое решение проблемы случайного перекрытия интерактивной области. Защита от случайного перекрытия предусматривает два уровня:

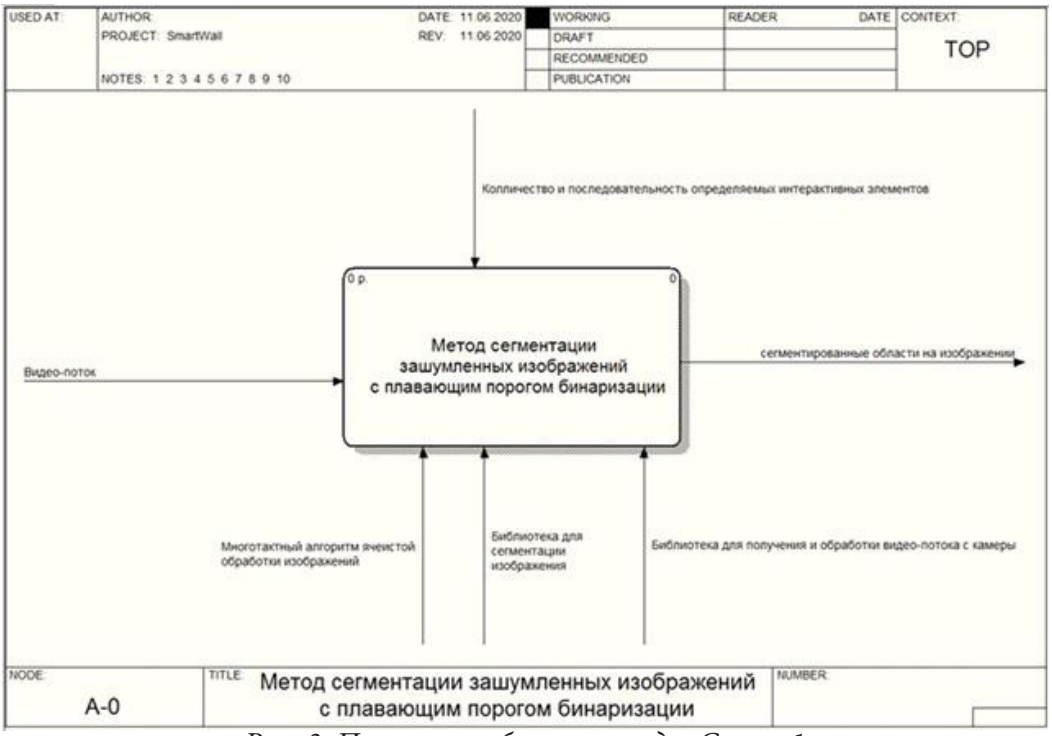


Рис. 3. Принцип работы метода. Схема 1

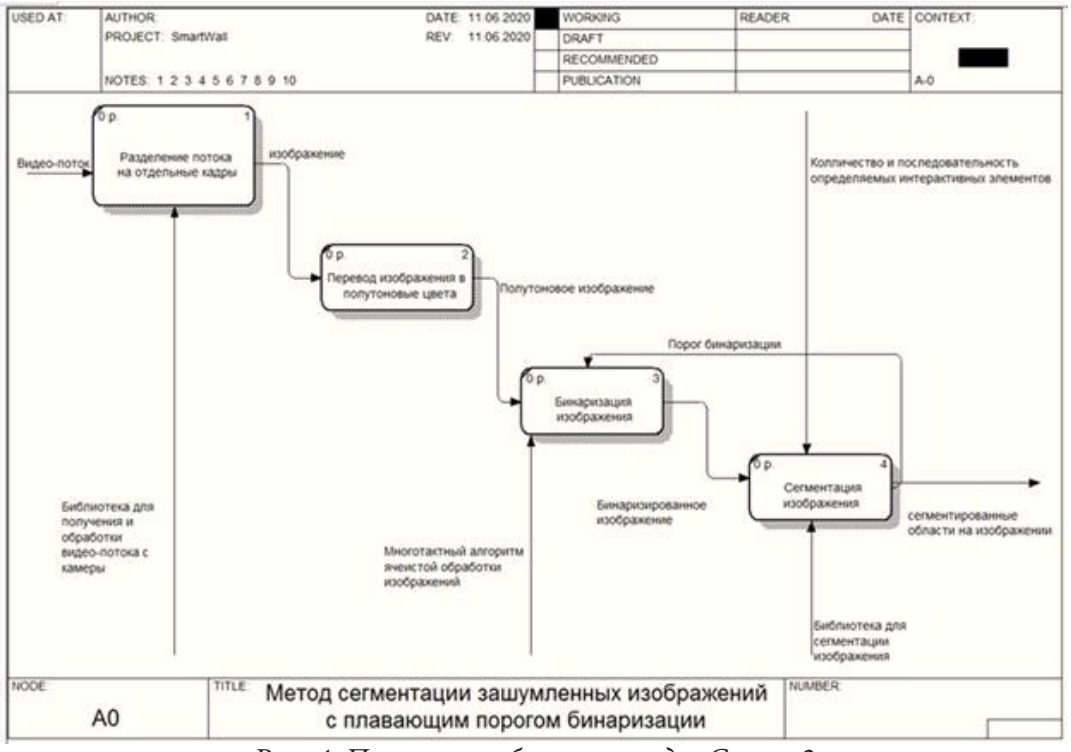


Рис. 4. Принцип работы метода. Схема 2

• таймингова защита – в коде программы предусмотрены специальные таймеры, которые позволяют компьютеру не осуществлять предусмотренное действие при случайном перекрытии интерактивной области;

• анатомическая защита – в коде программы алгоритмически предусмотрен вариант перекрытия нескольких интерактивных областей одновременно, при котором предусмотренное действие на компьютере совершается только по результату перекрытия одной области, которая выбирается с учетом анатомии человека.

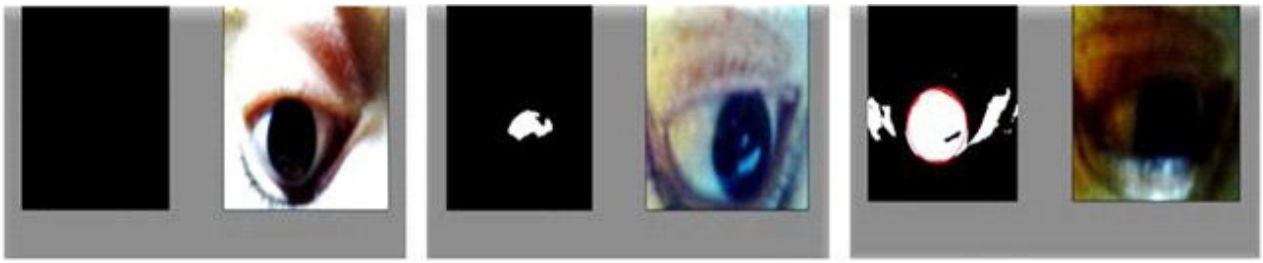


Рис. 5. Скриншоты программы по сегментации зрачка глаза

3.2. Разработка программных средств, обеспечивающих совместную работу веб-камеры и одноплатного компьютера Raspberry Pi

Программные средства, обеспечивающие совместную работу веб-камеры и одноплатного компьютера Raspberry Pi, были разработаны на языке программирования Java 8 [2, 8], в среде разработки IntelliJ IDEA, для ОС Linux. Разработанные программные средства предназначены для видео-захвата изображения с последующей обработкой по методу сегментации зашумленных изображений с плавающим порогом бинаризации.

В ходе разработки экспериментальным путем было установлено, что повышение ресурсоемкости процессов на одноплатном компьютере Raspberry Pi, приводит к его нагреванию и потере до 30 % мощностных характеристик, таких как частота процессора и скорость обработки данных, что ведет к соответствующему уменьшению быстродействия ПО SmartWall. Результатом эксперимента стало принятие решения о целесообразности схемы работы программного обеспечения, представленной на рис. 6.

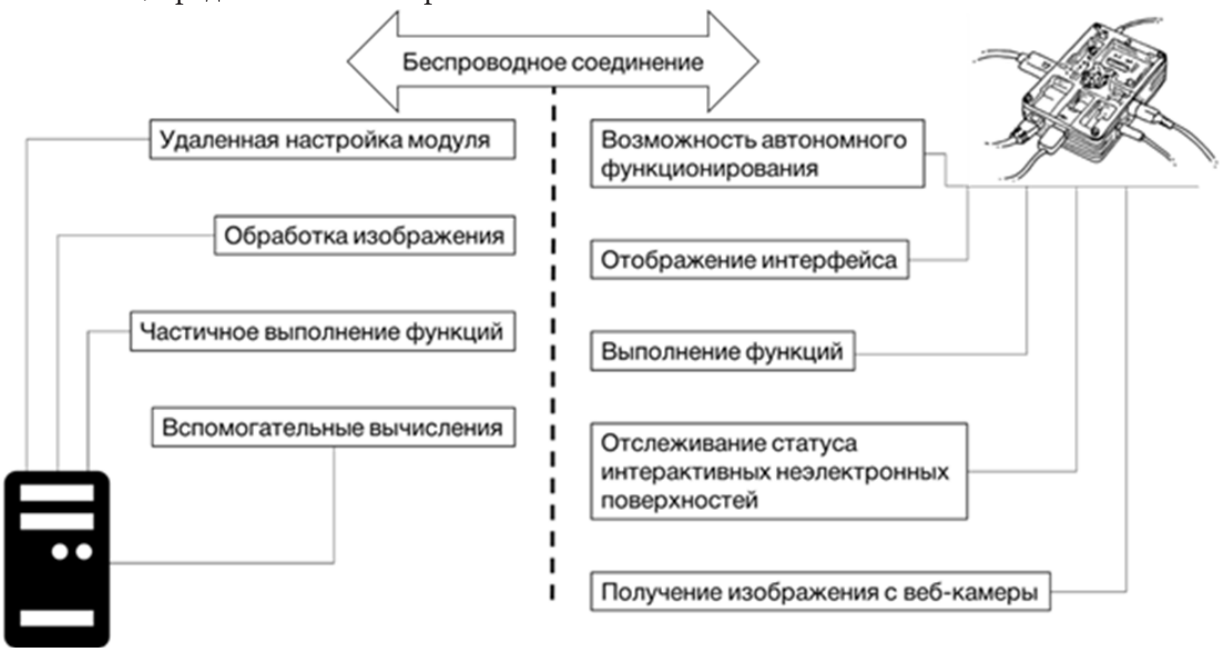


Рис. 6. Схема принципа работы модуля

4. Реализация аппаратно-технических задач

В ходе работы были решены конструктивные проблемы соединения веб-камеры с одноплатным компьютером Raspberry Pi для создания модуля удаленного доступа, а именно разработка оптимальной конструкции соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall.

Результаты тестирования показали, что пассивного охлаждения модуля недостаточно для его стабильной и продуктивной работы. По результатам тестирования было принято решение добавить в конструкцию модуля элемент активного охлаждения, а именно процессорный кулер для Raspberry PI 3.

Конструкция мобильного модуля SmartWall, оптимизированная по результатам тестирования, представлена на рис. 7.

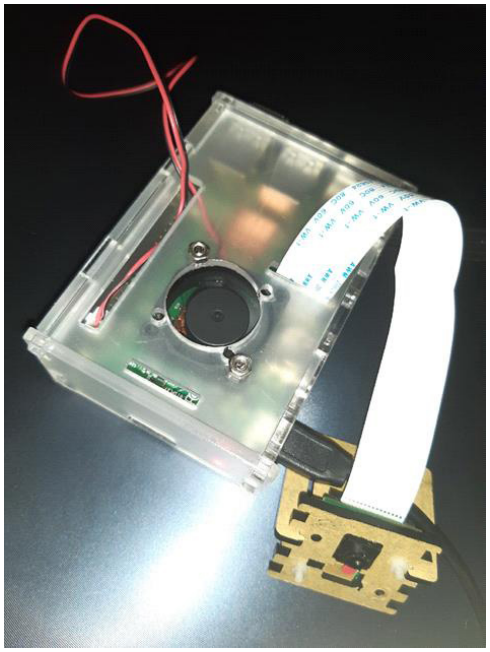


Рис. 7. Мобильный модуль SmartWall с системой активного охлаждения

Заключение

В ходе работы над проектом оптимизирован код программы, разработаны программные средства, обеспечивающие совместную работу веб-камеры и одноплатного компьютера Raspberry Pi, разработана конструкция соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall.

Опубликованы статьи, посвященные актуальным проблемам разработки [3, 4, 5, 6, 7]. Проект был представлен на различных научных форумах и конференциях, таких как Ежегодный саммит молодых ученых и инженеров «Большие вызовы для общества, государства, науки» (Сочи 2018, Сочи 2019), IV Всероссийский форум «Наука будущего — наука молодых» III Международной научной конференции «Наука будущего» (Сочи 2019), Falling Walls Lab Moscow 2019. Выполнен план НИР по проекту в рамках гранта от ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере» (Фонд содействия инновациям), полученного автором в качестве победителя программы УМНИК 2018 г. Получено три Свидетельства о государственной регистрации программ для ЭВМ [9, 10, 11].

Благодарности

Научный руководитель проекта И. Е. Воронина, д-р техн. наук, доц., профессор каф. программного обеспечения и администрирования информационных систем.

НИР по проекту проводится в рамках гранта от ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере» (Фонд содействия инновациям).

Литература

1. *Тропченко, А. Ю.* Методы вторичной обработки изображений и распознавания объектов. Учебное пособие / А. Ю. Тропченко – СПб. : СПбГУ ИТМО, 2012. – 52 с.

2. *Окулов, С. М.* Основы программирования / С. М. Окулов. – 5-е изд., испр. – М. : БИНОМ. Лаборатория знаний, 2010. – 440с. : ил.

3. *Принев, М. А.* Программный комплекс SmartWall для создания интерактивных поверхностей и гаджетов // Информатика: проблемы, методология, технологии: сб. матер. XVIII международной научной конференции: в 7 т. / под ред. Н.А. Тюкачева, Воронеж, Воронежский государственный университет. – Воронеж: Издательство «Научно-исследовательские публикации» (ООО «Вэлборн»), 2018. – Т.4. – С. 194–198.

4. *Принев, М. А.* Использование ПО SmartWall для решения проблем внедрения ИКТ в образовательную среду // Математика, информационные технологии, приложения: сб. науч. тр. / под ред. А.И. Шашкина; Воронежский государственный университет. – Воронеж : Издательский дом ВГУ, 2018. – С. 84–89.

5. *Принев, М. А.* Перспективы использования по SmartWall в образовательной среде / М. А. Принев // Вестник воронежского института высоких технологий. – 2019. – № 1 (28). – С. 96–99.

6. *Принев, М. А.* Оптимизация метода сегментации зашумленных изображений с плавающим порогом бинаризации // Современные проблемы прикладной математики и информатики [Текст]: сборник трудов научной сессии, Воронеж, 2019 г. / ФГБОУ ВО «Воронежский государственный университет». – Воронеж: Издательско-полиграфический центр «Научная книга», 2019. – С. 81–83.

7. *Принев, М. А.* Разработка программного обеспечения на основе метода сегментации зашумленных изображений с плавающим порогом бинаризации для создания интерактивных неэлектронных поверхностей и гаджетов / М. А. Принев // Тезисы докладов третьей Международной научной конференции «Наука будущего» и четвертого Всероссийского молодежного научного форума «Наука будущего наука молодых» – Сочи, 2019. – С. 10.

8. Документация по Java Platform Standard Edition 8. – URL: https://docs.oracle.com/javase/8/docs/ (дата обращения: 05.10.2020).

9. Свид. 2018611550 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. SmartWall / М. А. Принев; заявитель и правообладатель ФГБОУ ВО «Воронежский государственный университет» (RU). – №2017662835; заявл. 11.12.17; опубл. 02.02.18, Реестр программ для ЭВМ. – 1 с.

10. Свид. 2020618747 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. SmartWall / М. А. Принев; заявитель и правообладатель ФГБОУ ВО «Воронежский государственный университет» (RU). – №2020617762; заявл. 14.07.20; опубл. 04.09.20, Реестр программ для ЭВМ. – 1 с.

11. Свид. 2020618286 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. SmartWall / М. А. Принев; заявитель и правообладатель Принев М. А. (RU). – №2020617321; заявл. 02.07.20; опубл. 22.07.20, Реестр программ для ЭВМ. – 1 с.

Принёв Мечислав Александрович – студент 4-го курса кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: metshislav@rambler.ru

Воронина Ирина Евгеньевна (научный руководитель) – д-р техн. наук, доц., профессор кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: irina.voronina@gmail.com

УДК 004.8

ПОСТРОЕНИЕ МОДЕЛИ БИНАРНОЙ СЕГМЕНТАЦИИ В ЗАДАЧЕ ОЦЕНКИ ЗЕРНИСТОСТИ СТАЛИ

Н. С. Пугачев

Воронежский государственный университет

Введение

В наши дни многие компании прибегают к внедрению методов машинного обучения для решения повседневных задач. Как правило, такие задачи однообразны и занимают у экспертов достаточно много времени.

Одним из основных свойств отвечающих за качество стали является ее зернистость. Более того, все механические свойства стали, такие как ударная вязкость и предел усталости, зависят только от размеров зерен. Зерна стали хорошо видны в микроскоп, если посмотреть на срез. Зернистое строение любого металла называется его структурой.

Целью данной работы было на основе изображений срезов стали построить модель, проводящую бинарную классификацию пикселей изображения, принадлежащих зерну или границе зерен. Полученная сегментация описывает границы зерен, исходя из которой можно найти их площадь. Данную задачу осложняет то, что величина и форма зерен любого металла не являются однородными, границы между зернами не имеют фиксированной толщины. Помимо этого, сами зерна имеют небольшие пустоты, а после среза на них остаются следы пилы, по виду напоминающие границу.

Задача бинарной сегментации встречается не только в области производства металлов. Так же она встречается при обработке биомедицинских изображений или поисков аномальных объектов на однородных поверхностях.

Сложность решения таких задач заключается в том, что, как правило, данных достаточно мало, при этом к результату работы алгоритма предъявляются высокие требования по точности, так как оценивается именно площадь объектов, а не просто их наличие.

В последние несколько лет, совершен достаточно большой прорыв в области обучения сверточных нейронных сетей во многом это обусловлено тем, что появилась возможность обучать их на больших объемах данных, таких как ImageNet, кроме того возросла и вычислительная мощность что позволяет конструировать сети с большим количеством параметров. Однако типичной задачей таких сетей является классификация изображений целиком, в то время как для задачи бинарной сегментации требуется классифицировать каждый пиксель на принадлежность к объекту. Первая попытка выделить клеточную структуру на изображениях была описана в работе Ciresan D.C. “Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images”[1]. Его метод был основан на методе скользящего окна для прогнозирования метки класса каждого пикселя, используя локальную область (патч) вокруг этого пикселя. Однако, такая сеть имеет огромное количество параметров и работает медленно, так как запускается для каждого патча. Кроме того, при выборе размера патчей возникает дилемма локализации и размера патча. Большие патчи требуют большего количества сжимающих пулинг-слоев, что снижает точность локализации, в то время как выбор патчей маленького размера приводит к тому, что модель не учитывает контекст и точность падает в целом. Следующие архитектуры [2] отличались тем, что для финальной классификации использовались признаки, полученные не только на последнем сверточном слое, но и с предыдущих, благодаря этому модели имели возможность учесть контекст и дать достаточно неплохую локализацию. Однако такие методы вычислительно сложные, требуют большого количества данных.

Новым словом в решении задач бинарной сегментации стали так называемые полностью сверточные сети [3]. Суть такой архитектуры в том, что вместо использования операторов пулинга используются слои, обратные свертке. Отличительной особенностью такой архитектуры является то, что признаки высокого разрешения совмещаются с выходом классификатора. Кроме того, что такой метод исключает дилемму локализации и размера патча, работа такой сети гораздо быстрее.

1. Материалы и методы

1.1. Исходные данные

Как было упомянуто выше, целью этой работы является построение модели способной распознавать зерна на срезе стали. Задача, обсуждаемая в этом исследовании, была поставлена компанией НЛМК. Данные представляю собой изображения срезов и разметку к ним. Пример такого снимка можно увидеть на рис. 1.

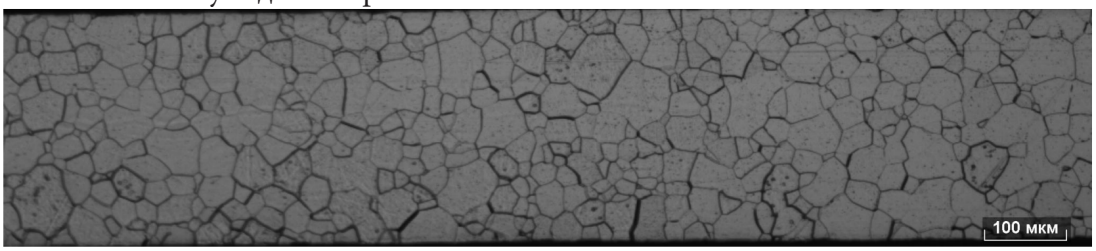


Рис. 1. Пример снимка среза стали

Как видно из рис. 1, количество зерен на одной фотографии достаточно велико, поэтому для упрощения работы и получения более точных результатов, было принято решения разбивать изображение на более мелкие части с перекрытием. Такой подход облегчает работу сети а перекрытие позволяет дважды прогонять через сеть участки изображения где граница не является явной.

После разработки инструмента нарезки изображения на патчи и восстановления из этих патчей исходного изображения, было принято решение расширить количество данных путем аугментации. Это связано с тем что, несмотря на то что архитектуры полностью сверточных сетей, не требуют огромного количества образцов для обучения, представленных картинок всё равно достаточно мало. Среди методов были использованы:

- приближение
- сдвиг
- поворот
- эластичная трансформация

Также в качестве первоначальной обработки данных использовалась нормализация путем вычитания среднего значения для каждого патча и деление его на среднеквадратичное (стандартное) отклонение. Ниже приведен пример данных после обработки и аугментаций.

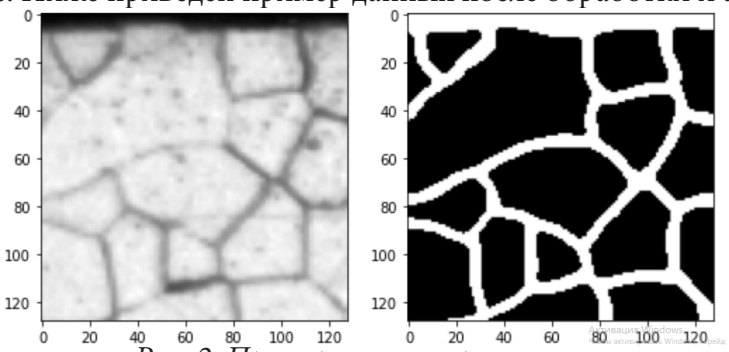


Рис. 2. Пример патча и разметки

1.2. Архитектуры полностью сверточных сетей

В данной работе были опробованы 3 архитектуры сетей: Unet[3], Unet++[4] и Wnet[5]. На рис. 3 представлена архитектура сети Unet.

Эта сеть отличается от обычных сетей классификаторов тем, что имеет часть, состоящую из операций, обратных свертке. Они содержат большое количество признаков в высоком разрешении, для того чтобы получать информацию о контексте со сверток, предшествующих классификации. Таким образом, сеть состоит из двух ветвей: энкодера и декодера.

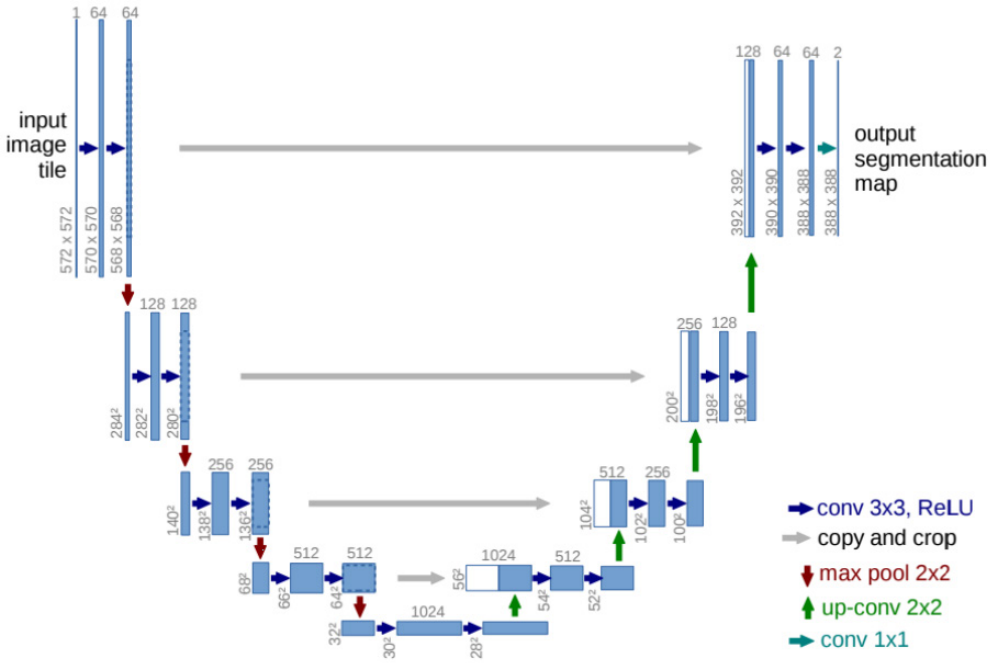


Рис. 3. Архитектура сети Unet

Сеть Unet++ ,как видно из названия, является модификацией сети Unet. Она также состоит из энкодера и декодера, но, в отличии от Unet, они связаны с помощью плотно связанных сверточных блоков. Суть серии таких блоков заключается в том, что карта признаков, извлеченная первым сверточным слоем передается на вход всех последующих сверточных слоев, как показано на рис. 4.

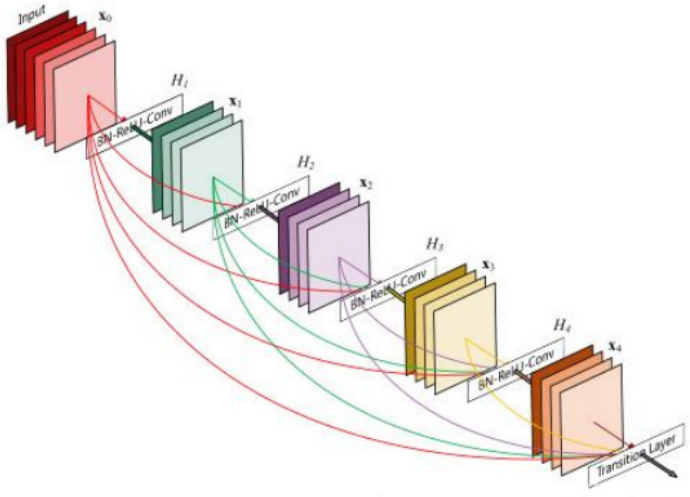


Рис. 4. Схема плотно связанных сверточных блоков

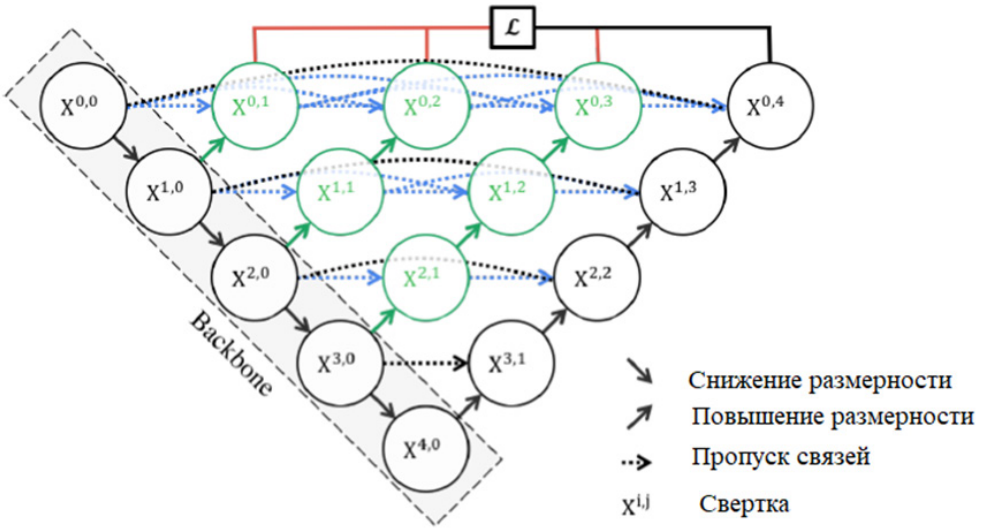


Рис. 5. Архитектура сети Unet++

На рис. 5 приведена схема сети Unet++. Основной причиной использования плотно связанных сверточных блоков является попытка преодоления семантического разрыва между картами признаков энкодера и декодера. На рис. 5 плотно связанные сверточные блоки обозначены зеленым цветом.

Другой модификацией Unet является сеть Wnet. Схема сети представлена на рис. 6. Как видно из рисунка, сеть представляет собой две сети Unet, причем на вход второй части подаётся как выход первой так и оригинальное изображение. В то время как целью архитектуры Unet++ является закрытие семантического разрыва между картами признаков энкодера и декодера, разработчики архитектуры Wnet ставят перед собой задачу дать сети возможность строить некоторую генерализацию по разным источникам данных(доменам). Это связано с тем, что входные данные могут различаться по уровню освещенности, яркости и контрастности. Из за этого признаки выученные на одном наборе могут быть бесполезны на другом. Данная проблема справедлива как для Unet так и для Unet++. В сети Wnet первая часть служит именно для решения этой задачи. Она приводит данные из разных источников к одному домену. А затем уже вторая ее часть делает предсказание опираясь не только на само изображение, но и на адаптированный результат работы первой.

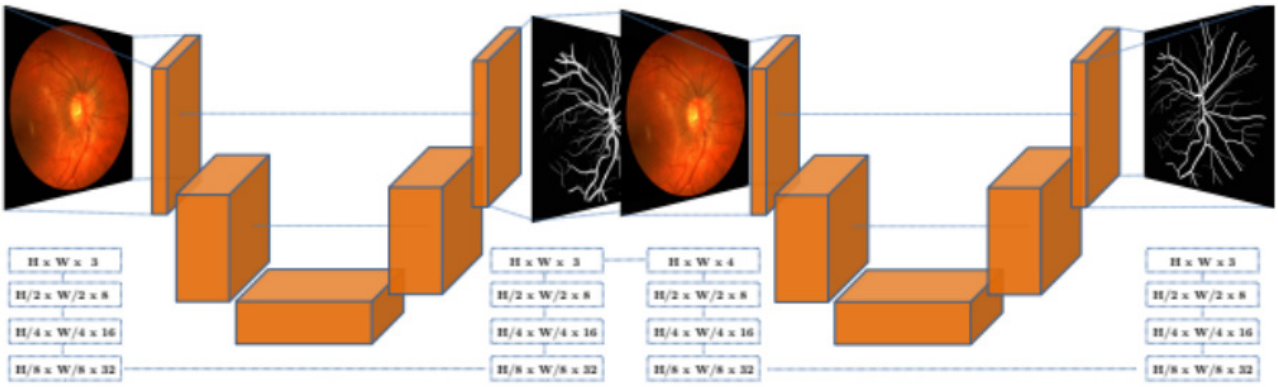


Рис. 6. Архитектура сети Wnet

2. Сравнение результатов

На рис. 7 приведены результаты работы трёх алгоритмов.

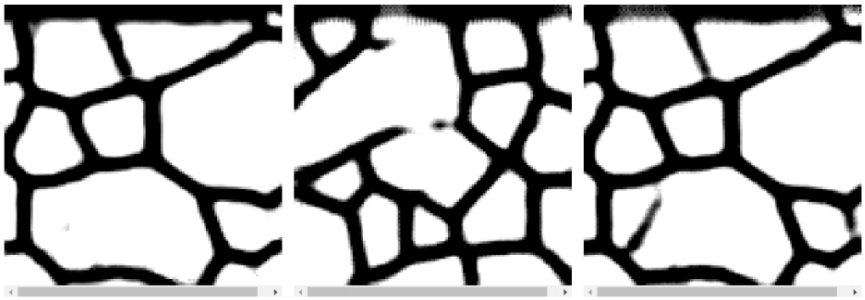


Рис. 7. Результаты работы сетей: Unet++, Unet и Wnet

Наилучшие результаты показала сеть Wnet. Unet++ показывает результаты лучше, чем Unet, но, из за небольшого количества данных, она склонна к переобучению. После прогона картинки через сеть патчи сшиваются обратно. На рис. 8 приведен пример результата работы сети Wnet.

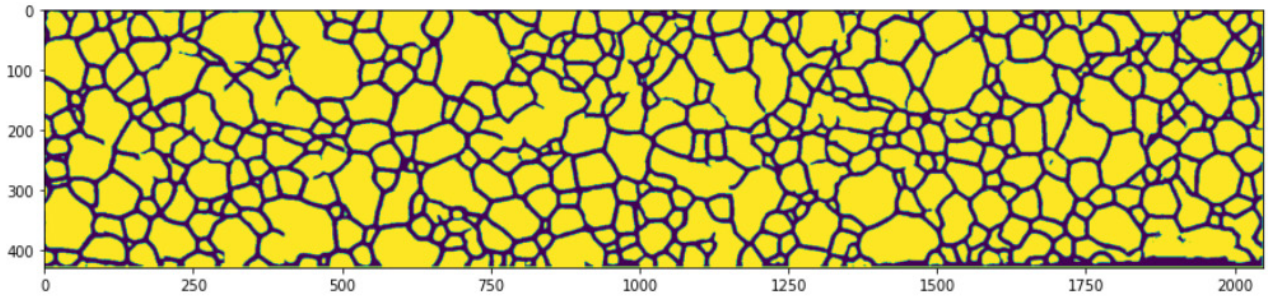


Рис. 8. Результат обработки

Далее на полученной маске находятся контуры зерен и строится гистограмма распределения их площадей. Ниже представлены графики оригинальных распределений одновременно с распределениями, полученными на выходе сетей:

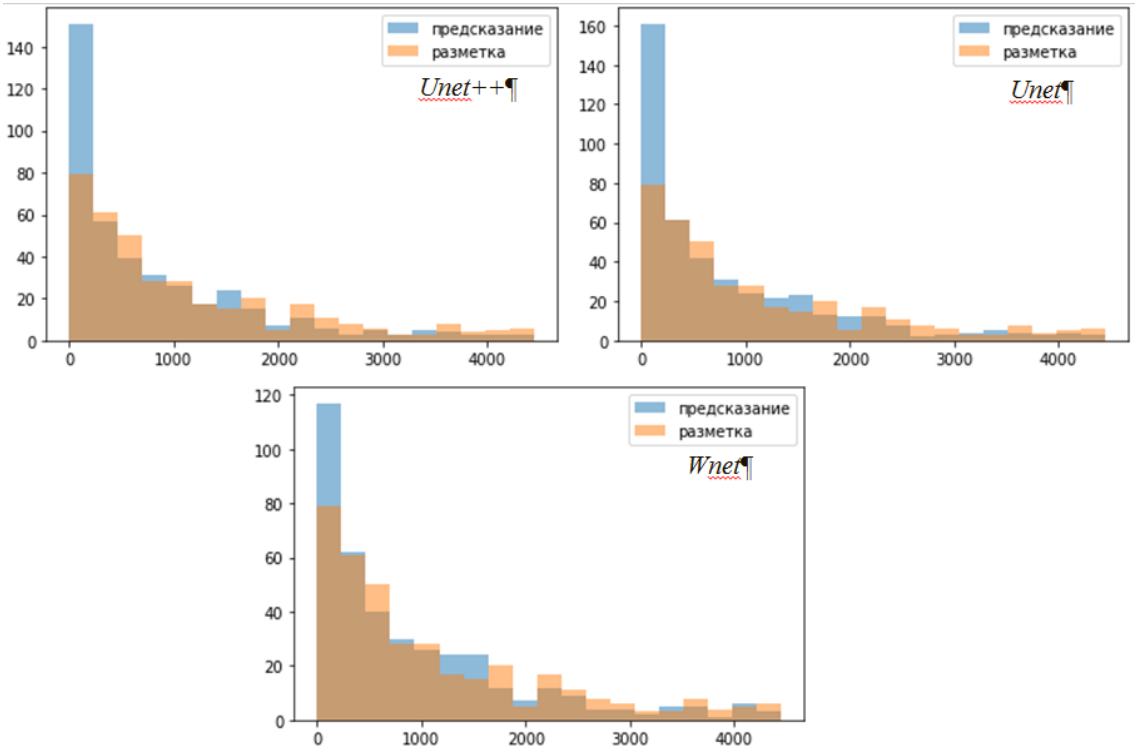


Рис. 9. Гистограммы площадей для Unet++, Unet и Wnet

Для сравнения гистограмм используется метрика корреляции гистограмм площадей. Результаты приведены в табл. 1.

Таблица 1

Сравнение гистограмм

Модель	Корреляция между гистограммами
unet	0.9108
Unet++	0.9017
Wnet	0.9591

Заключение

В результате данного исследования были изучены архитектуры трех сетей: Unet, Unet++ и Wnet. Реализованы алгоритмы препроцессинга, а именно разбиение изображения на патчи и эластичные трансформации. Описан цикл тренировки и подобраны оптимальные параметры для каждой сети. По итогам был сделан вывод о том, что результаты архитектуры сильно зависят от того насколько образцы входных данных похожи между собой (принадлежат к одному домену). Наилучшим образом показала себя сеть Wnet, которая с помощью первой части адаптирует образцы к некоторому единому виду который подается во вторую часть.

Литература

1. Deep neural networks segment neuronal membranes in electron microscopy images. / D. C. Ciresan, L. M. Gambardella, A. Giusti, J. Schmidhuber // ResearchGate. – 2021. – URL: https://www.researchgate.net/publication/267709055_Deep_Neural_Networks_Segment_Neuronal_Membranes_in_Electron_Microscopy_Images (дата обращения 20.01.2021)

2. Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks / M. Seyedhosseini, M. Sajjadi, T. Tasdizen // Computer Vision (ICCV), IEEE International Conference. – 2013. – с. 2175.

3. U-Net: Convolutional Networks for Biomedical Image Segmentation / Olaf Ronneberger, Philipp Fischer, Thomas Brox // Arxiv. – 2015. – URL: https://arxiv.org/pdf/1505.04597.pdf (дата обращения 20.05.2020)

4. UNet++: A Nested U-Net Architecture for Medical Image Segmentation / Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang // Arxiv. – 2018. – URL: https://arxiv.org/pdf/1807.10165.pdf

5. The little W-Net that could: state-of-the-art retinal vessel segmentation with minimalistic models / Adrian Galdran1, André Anjos, José Dolz, Hadi Chakor, Hervé Lombaert, Ismail Ben Ayed // Arxiv. – 2020. – URL: https://arxiv.org/pdf/2009.01907v1.pdf

Пугачев Николай Сергеевич – магистрант 2-го курса кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: pugachev1997@bk.ru

Каширина Ирина Леонидовна (научный руководитель) – д-р техн. наук, доц., профессор кафедры математических методов исследования операций Воронежского государственного университета. E-mail: kash.irina@mail.ru

УДК 004.4

РАЗРАБОТКА ПРОГРАММНЫХ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ АНАЛИЗА WEB-САЙТОВ

Н. Д. Пушки

Воронежский государственный университет

Введение

При разработке и модернизации программного обеспечения проводится анализ его структуры, связей между элементами и определение основных параметров функционирования. Такой анализ позволяет анализировать различные процессы и прогнозировать их развитие. В настоящее время данный подход широко используется при анализе программного обеспечения web-серверов и сетей, построенных на их основе. Структуру таких объектов наиболее часто описывают при помощи веб-графов.

Веб-граф – это структура данных, описывающая какой-либо веб-сайт или сеть. Основными элементами графа являются: вершины – страницы сайта, а ребра – гиперссылки, связывающие эти страницы.

Целью выполняемой работы является исследование устойчивости веб-графов, то есть определение насколько устойчивым является его структура к внешним воздействиям, таким, как выход вершины из активного состояния с последующей недоступностью всех его ребер, связывающие вершину с другими вершинами.

1. Постановка задачи

На данном этапе исследования требуется разработать программное обеспечение, позволяющее выполнять сбор и анализ большого количества данных для построения графа, состоящего из правильных вершин. В качестве основных задач можно выделить:

- Разработка приложения, реализующее механизм сбора данных с веб-сайтов.
- Сбор необходимых данных, с помощью приложения-сборщика.
- Разработка приложения, выполняющее построение матрицы смежности на основе собранных данных.
- Визуализация результатов исследования.

Основными объектами исследования будут являться внешние веб-сайты факультетов ПММ и ФКН.

Исследование, которое будет выполнено в данной работе, представляет собой общее изучение и оценку устойчивости указанных веб-сайтов к атакам посредством анализа зависимости размера гигантской компоненты сильной связности от шанса удаления вершины графа.

2. Определение используемых инструментов

Перед началом разработки приложения, реализующее механизм сбора данных обязательно было определить основные инструменты, с помощью которых происходит сбор данных со страниц. В настоящее время практически все сайты используют современные интернет-технологии, поэтому при реализации механизма нужно учитывать возможное наличие динамического контента, который будет генерироваться через обращение к API браузера. Данная проблема не позволит нам напрямую получать и парсить статический контент через GET за-

просы, так как часть контента может просто не отобразиться из-за отсутствия клиентского рендеринга. Для того, чтобы учитывать динамический контент - было решено использовать конкретный драйвер браузера Selenium Web Driver.

Использование драйвера дает возможность не только получить динамический контент, но и обращаться к API браузера через клиентский код. Данный механизм позволяет отказаться от явного парсинга страницы и выполнять поиск нужных гиперссылок с помощью CSS-селекторов. Таким образом, для сбора данных на одной странице, можно свести код приложения к построению верных селекторов и вызова нужного функционала браузера.

3. Классификация данных

Перед построением селекторов мною было выведено несколько основных правил, по выборке и классификации ссылок. При анализе гиперссылок основное внимание уделялось на содержимое атрибутов href и onclick. Ссылки были разбиты на 3 основные категории:

- Межстраничные внутренние ссылки.
- Внешние ссылки.
- Другие ссылки.

Межстраничные внутренние ссылки – гиперссылки, которые можно считать вершинами веб-графа текущего сайта. Эта группа ссылок ведет на веб-страницы внутри этого сайта. Ссылка не содержит встроенного в разметку клиентского кода на событие onclick, а также не содержит каких-либо идентификаторов в атрибуте href, намекающих на то, что это не прямая ссылка на другой ресурс.

Внешние ссылки – прямые ссылки, которые ведут на внешние ресурсы.

Другие ссылки – ссылки, как внешние, так и внутренние, которые не ведут на веб-страницы, которые можно считать вершинами графа.

В эту категорию в своем исследовании я относил следующие гиперссылки:

- Содержащие якорь #
- Содержащие клиентский код onclick
- Ссылки на медиафайлы (.png, .jpeg, .avi, .flv)
- Ссылки на документы (.pdf, .xlsx)
- Ссылки на архивы (.zip, .7z)
- Ссылки, инициирующие отправку электронной почты (:mailto)
- Ссылки на скрипты (.php, .jsp, .aspx)

4. Нормализация URL

Некоторые внутренние ссылки содержат параметры, и нуждаются в нормализации. Если обратиться к теории построение правильного API веб-приложения, также известного как RESTful API, то можно выделить некую закономерность. Параметры в любой гиперссылке начинают перечисляться после знака вопроса, разделяя друг друга знаком амперсанда “&”. В основном параметры используются для фильтрации, меняют отображаемый контент, но остаются одной единственной страницей. В моем исследовании, подобные страницы будут рассматриваться как одна единственная вершина и приводиться к адресу, который расположен до вопросительного знака.

5. Алгоритм работы программы

В качестве хранилища результата обхода мною используется легковесная реляционная база данных sqlite3. Она поддерживает все необходимые операции и позволит сохранять как промежуточные данные, необходимые во время обхода, так и результирующий сет данных.

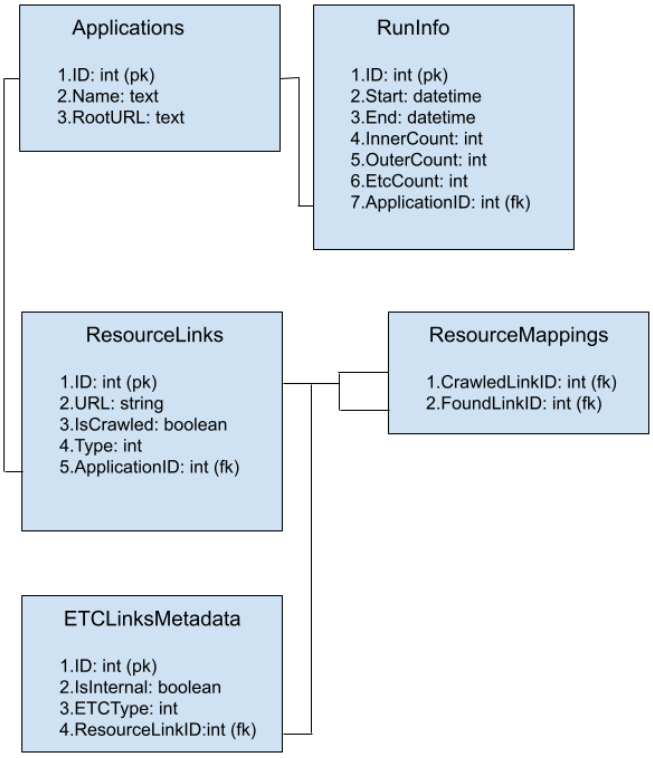


Рис. 1. Схема базы данных анализатора

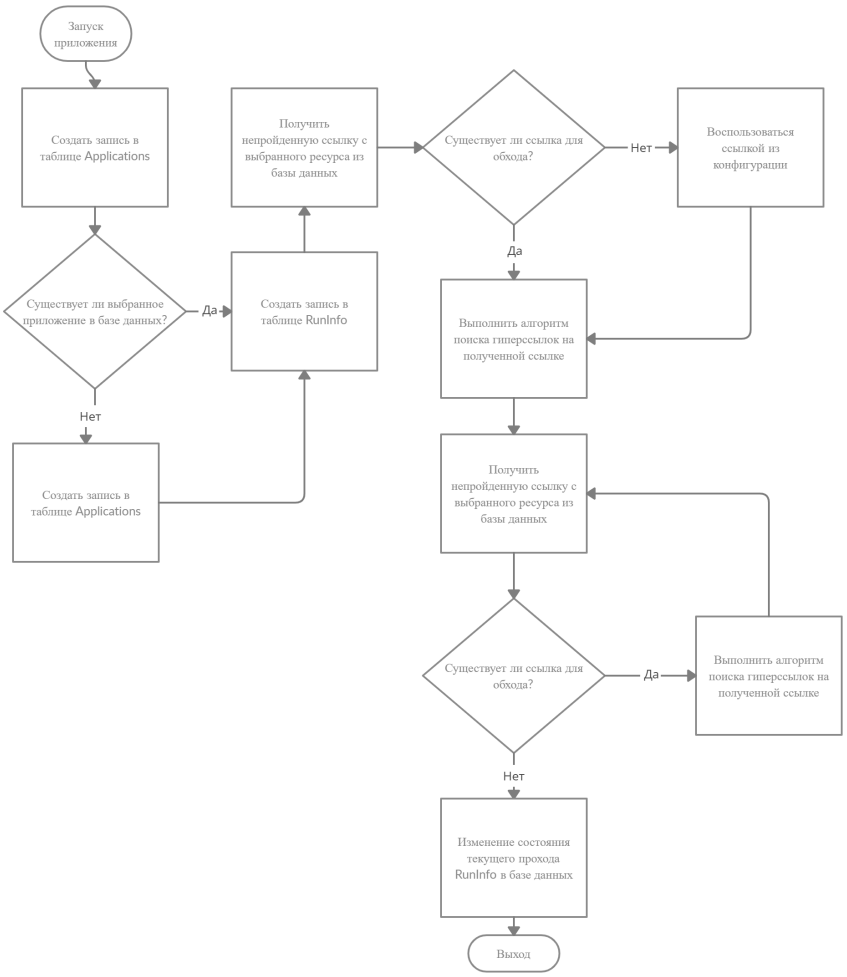


Рис. 2. Алгоритм общего обхода веб-сайта

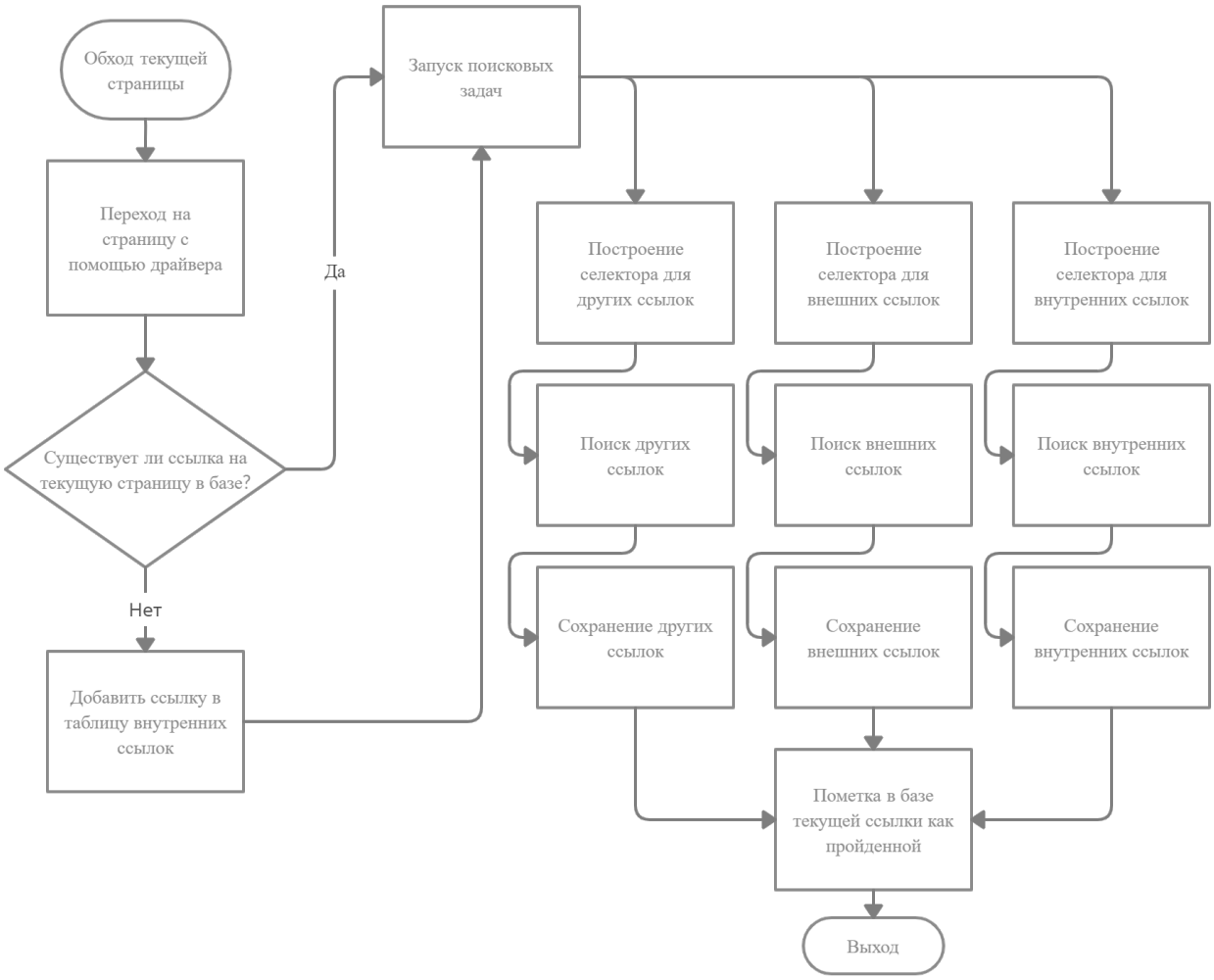


Рис. 3. Алгоритм поиска гиперссылок на странице

6. Результаты анализа

После выполнения алгоритма на указанных сайтах были получены следующие результаты:

Веб-Ресурс	Внутренние ссылки	Внешние ссылки	Другие ссылки
ПММ	637	151	5095
ФКН	936	396	867

Такая крупная разница в категории другие ссылки является следствием того, что на веб-сайте пмм есть мастер-страница содержит ссылки с якорями. Из-за этого, практически на каждой внутренней странице повторяется этот набор гиперссылок с одинаковым якорем, но разным URL.

Заключение

В рамках данной работы было разработано программное обеспечение для сканирования сайтов с целью построения их веб-графов, а также анализа полученного графа. Программное обеспечение было использовано на двух веб-сайтах факультетов ВГУ. Результатом данной работы можно считать полученный набор данных, который можно будет исследовать на устойчивость в рамках научной исследовательской работы.

Литература

1. *Björneborn L., Ingwersen P.* Toward a basic framework for webometrics // Journal of The American Society for Information Science and Technology. 2004. Vol 55(14). P. 1216-1227.

2. *Печников А. А.* Построение и исследование веб-графа информационного веб-пространства Санкт-Петербургского Государственного Университета // Фундаментальные исследования. – 2015. – № 10 (часть 3) – С. 512–517.

3. .NET Documentation - https://docs.microsoft.com/en-us/dotnet/

4. Сайт факультета ПММ - http://www.amm.vsu.ru/

5. Сайт факультета ФКН - https://www.cs.vsu.ru/

Пушкин Николай Дмитриевич – студент 2-го курса магистратуры кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: hitpoin1@gmail.com

Абрамов Геннадий Владимирович (научный руководитель) – д-р техн. наук, проф., зав. кафедрой математического обеспечения ЭВМ, профессор кафедры математического и прикладного анализа. E-mail: agwl@yandex.ru

УДК 517

УРАВНЕНИЕ ДЛЯ НАХОЖДЕНИЯ МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ

И. О. Русинова

Воронежский государственный университет

Введение

Математическое ожидание случайной величины – наиболее важная числовая характеристика случайной величины (вместе с дисперсией и моментами более высоких порядков). Если известна функция распределения, вычислить математическое ожидание несложно. Однако, важнее получить численные характеристики, например, решения дифференциального уравнения.

1. Постановка задачи

Рассмотрим задачу Коши для дифференциального уравнения второго порядка:

$$\ddot{x} + f(t)\dot{x} + \varepsilon(t)x = 0 \tag{1}$$

$$\begin{cases} x(t_0) = x_0 \\ \dot{x}(t_0) = x_1, \end{cases} \tag{2}$$

где $f(t)$ – детерминированный процесс, $\varepsilon(t)$ – случайный процесс, x_0, x_1 – независимые от ε случайные величины, $x : R \to R$ – искомый случайный процесс.

Требуется найти математическое ожидание решения задачи Коши.

2. Решение задачи

Предположим, что известны моменты первого порядка начальных условий, то есть величины $M[x_0]$, $M[x_1]$.

Пусть известен характеристический функционал:

$\varphi_\varepsilon(u) = M\left[\exp\left(i\int_{t_0}^{t_1}\varepsilon(s)u(s)ds\right)\right]$, где u – суммируемая функция на отрезке $[t_0, t_1]$.

Введем условное обозначение для упрощения записи:

$$w = \exp\left(i\int_{t_0}^{t_1}\varepsilon(s)u(s)ds\right).$$

2.1. Вспомогательное уравнение

Умножим уравнение на w и возьмем математическое ожидание, получим

$$M\left[\ddot{x}w\right] + M\left[f(t)\dot{x}w\right] + M\left[\varepsilon(t)xw\right] = 0. \tag{3}$$

Найти решение этого уравнения не представляется возможным. Поэтому введем вспомогательное отображение

$$y(t,u) = M\left[x(t)w\right].$$

Очевидно, что $y(t,0) = M[x(t)]$, а найти требуется математическое ожидание $M[x(t)]$.
Отметим, что

$$M[\ddot{x}w] = \frac{\partial^2 y(t,u)}{\partial t^2}$$

$$M[f(t)\dot{x}w] = f(t)\frac{\partial y(t,u)}{\partial t}.$$

Для нахождения математического ожидания случайной величины воспользуемся вариационной производной.

Определение. Пусть C – комплексная плоскость, $T = [t_0, t_1]$ – отрезок вещественной оси R, X – банахово пространство функций $x : T \to R$ с нормой $\| \ \|$ и L – подпространство в X. Будем полагать, что множество суммируемых с квадратом функций на T плотно в L.

Если для функционала $y : X \to C$ приращение $\Delta y(x) = y(x+h) - y(x)$ можно записать в виде $\Delta y(x) = \int_T \varphi(t,x)h(t)dt + \omega(x(\ ), h(\ ))$, $\forall h(\ ) \in L$, где интеграл понимается в смысле Лебега, $\varphi : R \times X \to C$, причем $\int_T \varphi(t,x)h(t)dt$ является линейным ограниченным по h функционалом на L и $\frac{|\omega(x(\ ), h(\ ))|}{\|h(\ )\|} \to 0$ при $\|h(\ )\| \to 0$, то $\varphi : T \times X \to C$ называется вариационной производной функционала y по направлению L и обозначается $\frac{\delta_L y(x(\ ))}{\delta x(t)}$.

Тогда

$$\frac{\delta \varphi_\varepsilon(u)}{\delta u(t)} = M\left[\exp\left(i\int_{t_0}^{t_1} \varepsilon(s)u(s)ds\right) i\varepsilon(t)\right]$$

$$\frac{\delta y(t,u)}{\delta u(t)} = M\left[x(t)wi\varepsilon(t)\right].$$

Уравнение (3) можно записать в виде:

$$\frac{\partial^2 y}{\partial t^2} + f(t)\frac{\partial y}{\partial t} + \frac{1}{i}\frac{\delta y}{\delta u(t)} = 0. \qquad (4)$$

Получили дифференциальное уравнение второго порядка с обыкновенной и вариационной производными, это уравнение не содержит случайных процессов.

2.2. Начальные условия

Произведем те же замены и преобразования с начальными условиями задачи Коши. Умножим равенства на w и возьмем математическое ожидание. Так как x_0, x_1 не зависят от ε, то получим:

$$M\left[x(t_0)w\right] = M\left[x_0 w\right] = M\left[x_0\right]M\left[w\right] = M\left[x_0\right]\varphi_\varepsilon(u)$$

$$M\left[\dot{x}(t_0)w\right] = M\left[x_1 w\right] = M\left[x_1\right]M\left[w\right] = M\left[x_1\right]\varphi_\varepsilon(u)$$

Таким образом, получаем начальные условия для уравнения (4):

$$y(t_0,u) = M\left[x_0\right]\varphi_\varepsilon(u)$$

$$\left.\frac{\partial y(t,u)}{\partial t}\right|_{t=t_0} = M\left[x_1\right]\varphi_\varepsilon(u).$$

Заключение

В итоге, получаем задачу Коши:

$$\frac{\partial^2 y}{\partial t^2} + f(t)\frac{\partial y}{\partial t} - i\frac{\delta y}{\delta u(t)} = 0$$

$$y(t_0, u) = M\left[x_0\right]\varphi_\varepsilon(u)$$

$$\left.\frac{\partial y(t,u)}{\partial t}\right|_{t=t_0} = M\left[x_1\right]\varphi_\varepsilon(u).$$

Решение полученной задачи можно найти численными методами. При подстановке $u = 0$ в решение $y(t,u)$, найдем математическое ожидание решения исходной задачи Коши (1), (2):

$$M\left[x(t)\right] = y(t,0).$$

Литература

1. Дифференциальные уравнения со случайными коэффициентами : учебное пособие для вузов / В. Г. Задорожний ; Воронежский государственный университет. – Воронеж : Издательско-полиграфический центр Воронежского государственного университета, 2012. – 98 с.

Русинова Ирина Олеговна – студент 4-го курса кафедры системного анализа и управления Воронежского государственного университета. E-mail: rusinova@amm.vsu.ru

Задорожний Владимир Григорьевич (научный руководитель) – д-р физ.-мат. наук, проф., заведующий кафедрой системного анализа и управления Воронежского государственного университета. E-mail: zador@amm.vsu.ru

УДК 004.8

ПОСТРОЕНИЕ МОДЕЛИ КЛАССИФИКАЦИИ ОБРАЩЕНИЙ В СЛУЖБУ ПОДДЕРЖКИ

А. Е. Саврасов

Воронежский государственный университет

Введение

В современном мире многие компании пытаются оптимизировать различные процессы своей деятельности с помощью алгоритмов машинного обучения исключением в данном тренде не являются и службы поддержки пользователей.

Автоматизация процесса адресации сообщения пользователей по соответствующим отделам поддержки позволит различным компаниям сократить время обработки запросов, поступающих к ним. Так же это позволит компаниям разгрузить свою службу поддержки, тем самым увеличить эффективность её работы.

Цель данной работы: взяв за основу информацию об обращении пользователей в службу поддержки (набор для обучения) с уже обозначенными отделами, которым данные обращения были отправлены, необходимо разработать модель машинного обучения, которая, на основе сообщения пользователя, будет предсказывать отдел, в который данное обращение нужно отправить.

В данной работе для классификации были исследованы и применены такие методы машинного обучения как LightGBM, метод опорных векторов, BERT. Для преобразования текстовой информации использовались следующие методы: Sentence-BERT, TF-IDF, UMAP.

В настоящее время исследованиям задач векторного представления текстов и их классификации уделяется большое внимание в зарубежных научных работах. В них можно увидеть эволюцию векторного представления текстовой информации. От представления текста на основе частотных характеристик слов в виде TF-IDF векторов или использования модели Word2Vec для сохранения семантической информации слов, до векторного представления, которое использует BERT. Векторное представление текстовой информации в BERT помимо семантической информации о слове, так же несёт информацию о контексте, в котором слово находится.

В зарубежных работах часто проводится сравнение методов преобразования текста в числовой формат. В данных работах полученные векторы признаков используют для обучения различных моделей машинного обучения с последующим сравнением метрик.

Одной из особенностей данной работы является наличие большой разницы между количеством документов в различных классах. Данная проблема так же является крайне важной исследовательской темой. Например, ученые из Университета Южной Калифорнии в своей работе исследовали различные подходы к решению данной проблемы. От аугментации векторного представления, до использования модели Sentence-BERT [1].

1. Материалы и методы

1.1. Исходные данные

Исходные данные были разделены на тренировочную и тестовую выборки. Тренировочная выборка включает в себя 21432 образцов, а тестовая выборка содержит 5358 образцов, разделенных на 78 классов. На следующих рисунках приведено облако наиболее встречающихся слов из текстов, принадлежащих шестому классу.

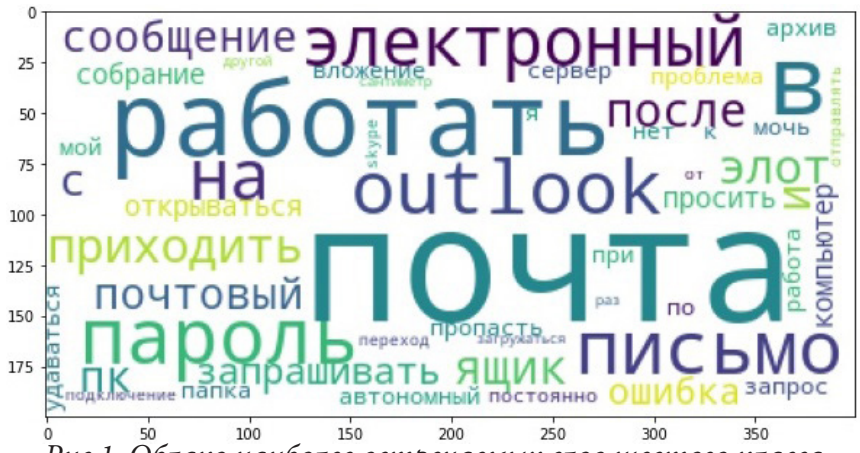


Рис 1. Облако наиболее встречаемых слов шестого класса

На рис. 2 показано распределение количества документов по классам. Можно заметить, что для некоторых классов количество обучающих примеров крайне ограничено, что в свою очередь усложняет обучение алгоритма классификации.

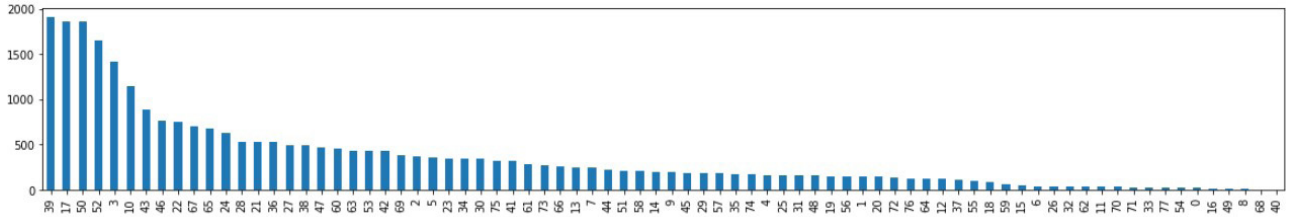


Рис. 2. Распределение количества документов по классам до удаления

По данной причине было принято решение по удалению двадцати классов, у которых количество документов было меньше ста тридцати экземпляров. Данные двадцать классов составляли лишь три процента от общего объема исходных данных. На рис. 3 показано распределение обращений по классам после удаления.

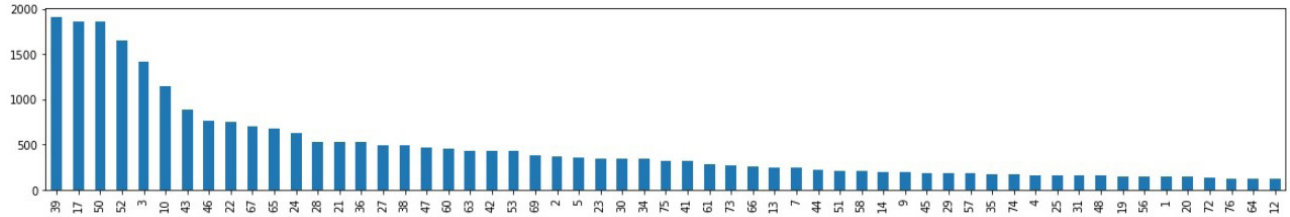


Рис. 3. Распределение количества документов по классам после удаления

Для решения поставленной задачи было выбрано три направления работы:

1. Использование TF-IDF как метода векторного представление текста. В последующем данные, полученные в результате такого представления были использованы для обучения метода опорных векторов.

2. Использование модели BERT с предварительно обученными параметрами и с последующей точной настройкой на представленной задаче. Полученная модель на этапе точной настройки будет использована для классификации.

3. Преобразование проблемы в задачу бинарной классификации. На получившейся задаче обучить Sentence-BERT. Преобразовать текст из исходной задачи в векторное представление с помощью, обученной BERT модели и применить метод опорных векторов или LightGBM для классификации получившихся векторов.

Так же для визуализации полученных векторных представлений был использован алгоритм UMAP. Пример подобной визуализации показан на рис. 4.

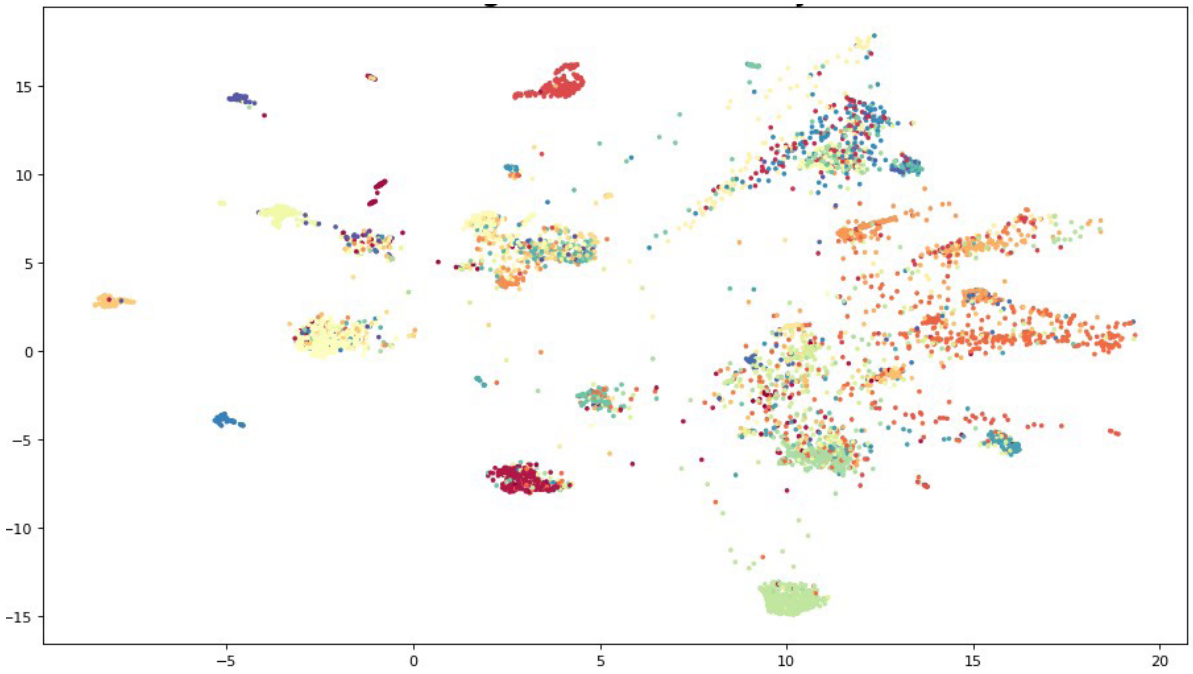


Рис. 4. Визуализация векторного представления с помощью UMAP

Данная визуализация была построена на основе векторного представления, полученного с помощью третьего метода. Легко заметить, что некоторые классы накладываются друг на друга. Это можно объяснить тем, что модель не может построить достаточно адекватное векторное представление для данных документов из-за сильной схожести документов некоторых классов.

1.2. Архитектуры используемых моделей

В первом методе используется TF-IDF для преобразования текстовых данных в векторные представление. В последующем данное векторное представление используется для обучения метода опорных векторов.

TF (term frequency – частота слова) – отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова t_i в пределах отдельного документа.

IDF (inverse document frequency – обратная частота документа) – инверсия частоты, с которой некоторое слово встречается в других документах коллекции. Основоположником данной концепции является Карен Спарк Джонс. Учёт *IDF* уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение *IDF*.

Метод опорных векторов в свою очередь обучается через поиск гиперплоскости, которая будет разделять точки в пространстве, которые принадлежат различным классам, с максимальным зазором. Под зазором здесь понимается сумма расстояний от плоскости до ближайшей точки по обе стороны.

Во втором методе будет использована нейросетевая модель BERT, основанная на архитектуре трансформер, для классификации и преобразования текстовых данных. BERT обучается методом стохастического градиентного спуска на основе входных данных.

Модель BERT изначально разрабатывалась для обработки последовательностей, таких как текст на естественном языке. BERT не требует обработки текста строго по порядку следования в документе. Благодаря этому BERT легко распараллеливается и может быть быстро обучена.

Во время обучения модель BERT учится находить векторное представления для каждого документа, представленного во входном наборе данных. В последующем данное векторное представление отправляется на вход уровня классификации. Уровнем классификации в данном методе является нейронная сеть прямого распространения.

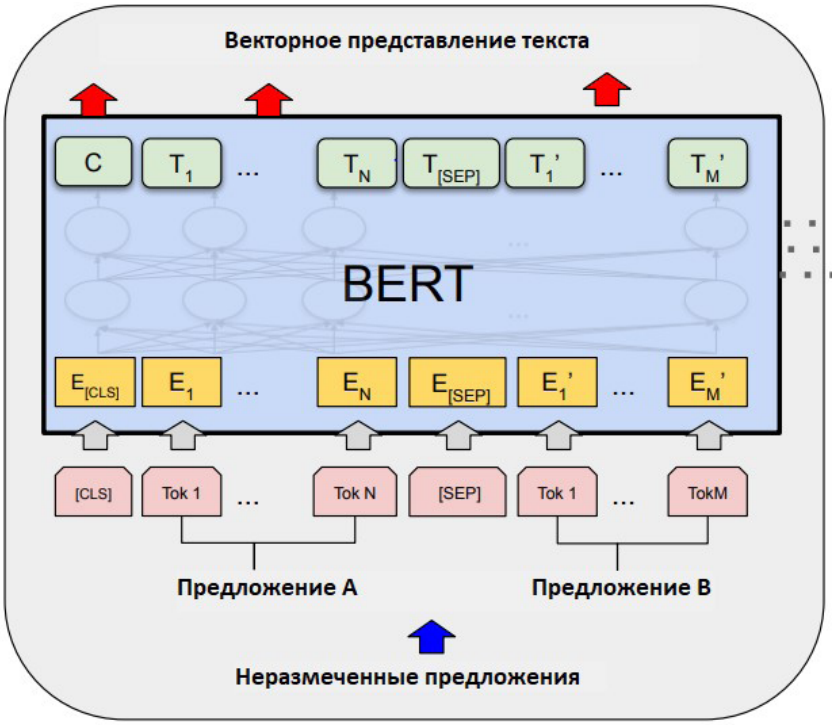


Рис. 5. Архитектура BERT [3]

Архитектура используемой модели показана на рис. 5. В BERT точная настройка выполняется путем простой замены соответствующих входных и выходных данных в зависимости от того, включают ли последующие задачи один текст или текстовые пары. Для каждой конкретной задачи подключают входные и выходные данные соответственно и полностью настраивают модель. Благодаря этому, BERT позволяет одной и той же предварительно обученной модели успешно решать самые разные задачи НЛП, такие как классификация текста, обнаружение сходства и многие другие.

Таблица 1

Влияние гипер-параметров на BERT accuracy [3]

Гипер-параметры				Accuracy на тестовых данных		
$\#L$	$\#H$	$\#A$	$LM(ppl)$	$MNLI-m$	$MRPC$	$SST-2$
3	768	12	5.84	77.9	79.9	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

В табл. 1 представлены результаты обучения модели BERT поставленной задачи в зависимости от различных значений гиперпараметров модели. Здесь $\#L$ – это количество слоев; $\#H$ – скрытый размер; $\#A$ – количество модулей внимания. $LM(ppl)$ – это степень неопределенности маскированной языковой модели LM удерживаемых обучающих данных.

В данном методе задача многоклассовой классификации была заменена на задачу бинарной классификации, в ходе решения которой определялось –принадлежат ли 2 документа одному классу или нет. И для данной задачи была обучена сиамская нейронная сеть, в которой базовой моделью является BERT. В зарубежной литературе модель, подобного вида называется Sentence-BERT [2]. Мотивацией для использования данной модели было желание получить наиболее точное векторные представления текстов, представленных в исходной задаче.

Сиамская нейронная сеть (англ. Siamese neural network) – это разновидность искусственной нейронной сети, которая состоит из двух идентичных нейронных подсетей с одинаковыми наборами весов. Данный вид сетей позволяет сравнить вектора признаков двух объектов с целью выделить их семантическое сходство или различие.

Существует три главных вида структуры сиамских нейронных сетей (рис. 6).

1. Входы x_i и x_j подаются на две параллельные подсети, представляющие собой две отдельные сети с одними и теми же весами, и смещениями. Для выходов h_i и h_j этих сетей подсчитывается метрика расстояния $d(h_i, h_j)$, которая подается на выходной слой, оценивающий схожесть $o(W)$ между входами подсетей.

2. Несколько последних слоев подсетей объединены, за ними следуют несколько дополнительных слоев. На последнем слое применяется softmax-преобразование. Данная архитектура известна под названием In-network stacking.

3. Оба входа x_i и x_j конкатенируются и подаются на вход единой сети, завершающейся слоем с softmax-преобразованием.

Для данной работы была использована модель типа 1. После обучения на задаче бинарной классификации BERT был использован для создания векторного представления текста из оригинальной задачи. Данное представление в свою очередь было использовано для обучения метода опорных векторов.

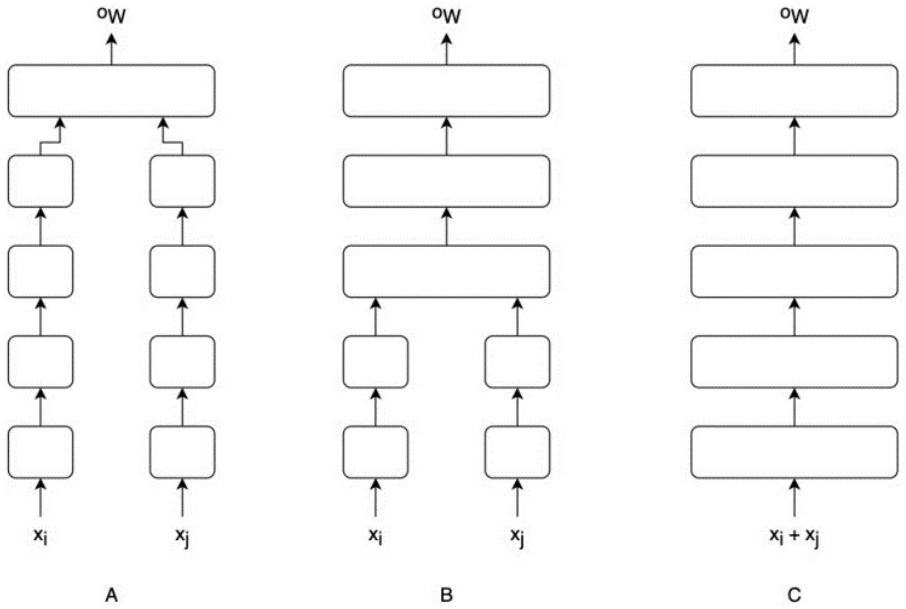


Рис. 6. Структуры сиамских нейронных сетей

2. Сравнение результатов

Для оценки результатов обучения использовалась метрика Accuracy (доля правильных ответов алгоритма).

В табл. 2 приведены средние результаты для всех трёх методов. В первом методе для векторного представления кроме отдельных слов были использованы ещё биграммы и триграм-

мы. Результат полученный первым методом крайне неплохой, учитывая, как мало примеров имеют некоторые классы. Так же стоит упомянуть, что для данного метода использовались все 78 классов.

Из таблицы так же видно, что наилучшие результаты в данной работе были получены с помощью BERT модели. Sentence-BERT в свою очередь показал худшие результаты, из чего можно сделать вывод, что полученные с помощью этой модели векторные представления не являются оптимальными для задач классификации.

Таблица 2

Результаты моделей

Название модели	Средняя точность
TF-IDF с SVM	67.4
BERT	72.9
Sentence-BERT с SVM	66.4

Заключение

Как видно из полученных результатов, наиболее успешным методом оказалась BERT модель с точной настройкой. Данный результат соотносится с результатами многих зарубежных исследований в которых производились сравнения различных векторных представлений для текстовой информации. В последующем планируется продолжать работать именно с этой моделью и улучшить результат с помощью подбора оптимальных гипер-параметров модели.

Литература

1. *Spangher, A.* Multitask Learning for Class-Imbalanced Discourse Classification / A. Spangher, J. May, L. May, S. Shiang // Arxiv. –2021. – URL: https://arxiv.org/abs/2101.00389v1 (дата обращения 20.01.2021)

2. *Reimers, N.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks / N. Reimers, I. Gurevych. // EMNLP. –2019. – URL: https://arxiv.org/abs/1908.10084 (дата обращения 01.02.2021)

3. *Devlin, J.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin, M. Chang, K. Lee, K. Toutanova // Arxiv. –2018. – URL: https://arxiv.org/abs/1810.04805 (дата обращения 20.01.2021)

4. *Жерон, О.* Прикладное машинное обучение с помощью Scikit-learn и TensorFlow / О. Жерон. – Севастополь : O'Reilly Media, 2012. – 308 с.

5. *Мюллер, А.* Введение в машинное обучение с помощью Python /А. Мюллер, С. Гвидо. – Филадельфия : J. Benjamins Publishing, 2010. – 393 с.

Саврасов Александр Евгеньевич – студент 2-го курса магистратуры кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: savrasov@amm.vsu.ru

Каширина Ирина Леонидовна (научный руководитель) – д-р техн. наук, доц., профессор кафедры математических методов исследования операций Воронежского государственного университета. E-mail: kash.irina@mail.ru

УДК 004.9

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ УПРАВЛЕНИЯ НАЛОГАМИ

А. Ю. Сапронов

Воронежский государственный университет

Введение

С каждым годом в мире появляется всё больше и больше новых решений, технологий, которые делают жизнь человека проще, удобнее, безопаснее. В период пандемии особенно ускорилось внедрение цифровых технологий. Однако, совершенствование технологий в разных сферах происходит с разной скоростью. Так, в налоговой системе до сих пор существует большое количество бизнес-процессов, которые требуют согласования между различными учреждениями, формирования сложных отчётов, подписания большой стопки бумаг. Но всё это можно упростить, сделав основную часть процессов автоматическими.

Главная сложность автоматизации состоит в том, что при составлении отчётов, договоров нам важно быть уверенными в том, что все условия договора были соблюдены, все данные были предоставлены верно. При увеличении количества участников (в налоговой системе их большое количество) это делать становится сложнее, и вероятность ошибки или фальсификации данных растёт.

Но уже сейчас существует технология, которая сможет гарантировать соблюдение всех условия договора, исключить подмену данных, избавиться от «третьей стороны». Это технология называется Блокчейн (англ. *Blockchain*). На Всемирном экономическом форуме в Давосе в 2016 году у 800 наблюдателей и технических специалистов спросили мнение о том, когда правительства начнут собирать налоги с помощью блокчейна. 73,1 % респондентов заявили, что это произойдёт в 2023–2025 годах [1].

1. Blockchain

1.1. Что такое Блокчейн

Блокчейн – распределенная база данных. Она хранит список упорядоченных записей, которые называются блоками. Каждый блок хранит информацию обо всех транзакциях, выполненных в определенный период, а также ссылку на предыдущий блок.

Главные преимущества использования блокчейна – это прозрачность проводимых транзакций и множественное копирование всех этих транзакций таким образом, что у каждого участника процесса всегда есть информация о каждом шаге всех партнеров [2]. Любой пользователь системы может просмотреть всю историю транзакций. Однако он не может идентифицировать получателя или отправителя информации, если не знает его уникальный номер. Но даже если пользователь узнает этот идентификатор, ему понадобится уникальный ключ доступf, чтобы совершить транзакцию от лица другого участника.

Ещё одна важная особенность блокчейна – компромисс. Любые данные, которые будут добавляться в систему, проверяются другими участниками. Если в итоге проверки найдутся ошибки, новая информация не будет записана в систему.

Допустим, один из пользователей решил удалить транзакцию, которую совершил ранее. Удалив транзакцию, он изменил свою копию базы данных и отправил её другим участникам, чтобы записать новую информацию в сеть. Другие пользователи сверяют новые данные со

своей копией. Они видят, что в полученных данных отсутствует транзакция, совершённая ранее и не принимают эти изменения. Это позволяет избежать подмены данных.

Пример совершения транзакции представлен на рис. 1.

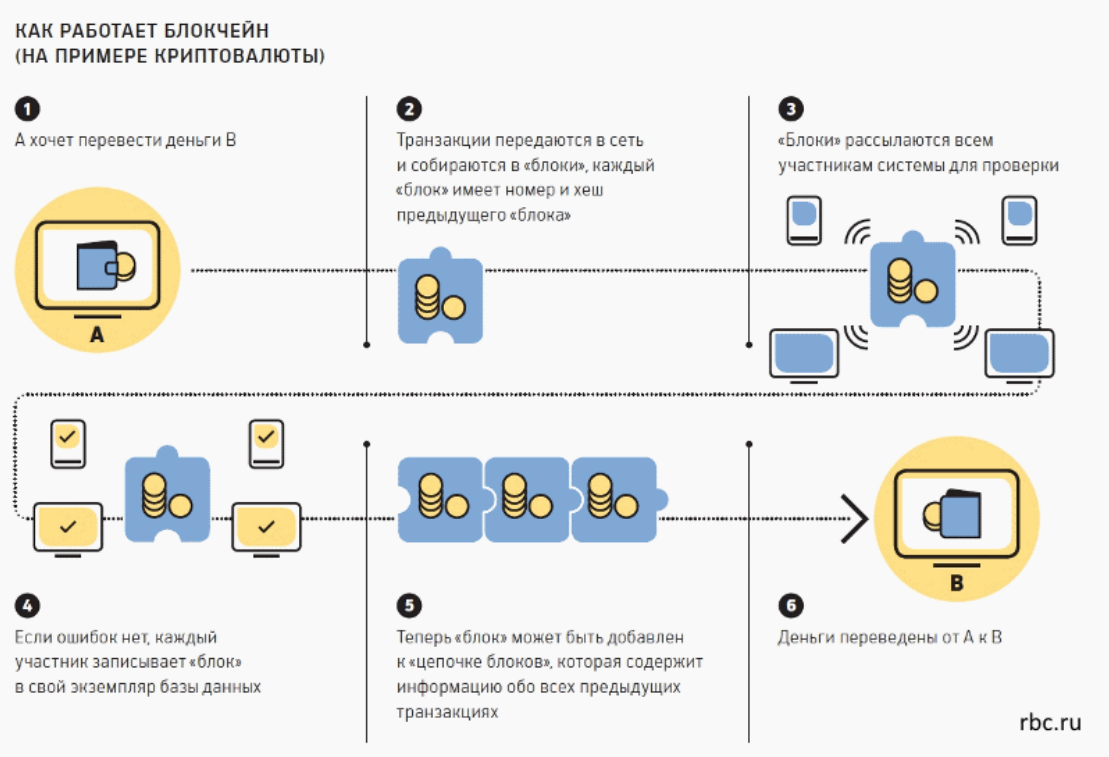


Рис. 1. Транзакция в блокчейне

1.2. Смарт-контракты

Изначально блокчейн содержал только информацию о транзакциях. Через некоторое время была создана блокчейн-платформа Ethereum, в которой можно было устанавливать условия для совершения транзакции. Эти условия можно было описывать в смарт-контракте.

Смарт-контракт – соглашение, в виде компьютерного кода. Код используется для ввода всех условий договора, заключенного между сторонами сделки. Обязательства участников предоставляются в контракте в форме «если-то» («если сторона А переводит деньги, тогда сторона В передает права на квартиру»). В контракте могут быть прописаны более двух участников. Они могут быть как отдельными лицами, так и организациями. Транзакция будет выполнена только тогда, когда все условия контракта будут выполнены. Этим гарантируется, что прописанное соглашение будет соблюдаться [3].

В итоге, используя смарт-контракты, вместо обычных договоров, мы получаем:

1. Более быстрое выполнение контракта. Нам не требуется вручную обрабатывать документ.
2. Независимость и экономию. Избавляясь от посредников, мы исключаем их вмешательство и сокращаем свои расходы.
3. Отсутствие ошибок. Автоматический расчёт убирает фактор ошибки в данных

2. Блокчейн в системе управления налогами

2.1. Блокчейн для работодателя

При выплате зарплаты работодатель обязан рассчитать сумму налога и перечислить его в бюджет. Помимо НДФЛ, на каждого сотрудника требуется делать взносы в фонды. Так в ПФР

начисляется 22 % от з/п, в ФСС 2,9 %, в ФФОМС 5,1 %. Получается работодателю нужно провести расчеты по сотрудникам, отправить взносы и отчётности в ФНС, ПФР, ФСС, Росстат [4]. Всё это занимает большое количество времени, расчётов. Также есть вероятность ошибки, которая может вскрыться на этапе налоговой проверки, что приведёт к штрафам.

Но этого можно избежать, внедрив блокчейн в систему уплаты налогов. Теперь работодатель просто выбирает сотрудника, вводит заработную плату и нажимает кнопку «Выплатить», об остальном позаботиться блокчейн.

После нажатия на кнопку в системе создаётся новая транзакция со смарт-контрактом. Этот контракт самостоятельно рассчитывает налог, отправляет нужную сумму в нужное налоговое учреждение, формирует требуемый отчёт и переводит деньги на счёт сотрудника. Тем самым процесс упрощается и становится более надёжным и дешёвым.

2.2. Блокчейн для сотрудника

Для сотрудника компании появится возможность видеть, сколько всего налогов за него уплатила компания и куда эти налоги пойдут. Благодаря этому у граждан будет расти уровень сознательности. Ведь многие просто не замечают сколько денег они отдают государству и на что они направляются.

В будущем можно дать людям самим выбирать то, куда отправить их налоги. И в этом случае блокчейн будет играть важнейшую роль.

3. Реализация смарт-контракта

Для написания смарт-контракта был выбран блокчейн Ethereum, так как он является самым популярным блокчейном, предназначенным для этого.

Задача смарт-контракта состоит в том, чтобы он сам высчитывал требуемые налоговые отчисления и переводил их в нужные учреждения. Для этого работодателю нужно будет ввести сумму выплачиваемой зарплаты и кошелёк своего сотрудника. Так же, будет необходимо иметь специальные права, чтобы устанавливать номер кошельков налоговых учреждений. Чтобы не писать много кода в одном контракте, мы разделим его на несколько.

3.1. Контракт Owner

В первую очередь создадим контракт Owner, который будет устанавливать владельца данного контракта. В листинге 1 представлен исходный код этого контракта.

Листинг 1

```
pragma solidity 0.8.4;

contract Owner {
    address private owner;

    event OwnerSet(address indexed oldOwner, address indexed newOwner);

    modifier isOwner() {
        require(msg.sender == owner, «Caller is not owner»);
        _;
    }

    constructor() {
        owner = msg.sender;
```

```
        emit OwnerSet(address(0), owner);
    }

    function changeOwner(address newOwner) public isOwner {
        emit OwnerSet(owner, newOwner);
        owner = newOwner;
    }

    function getOwner() external view returns(address) {
        return owner;
    }
}
```

В этом контракте есть приватное поле owner, в которое, при развертывании, помещается адрес кошелька человека, публикующего смарт-контракт в блокчейн. Так же есть модификатор isOwner, который проверяет, является ли владельцем, кошелёк взаимодействующий с контрактом.

3.2. Контракт TaxInstitutionWallets

Далее нам потребуется контракт, который будет хранить адреса кошельков налоговых учреждений и давать возможно их изменять. В нашем случае будет 4 кошелька: НДФЛ, ПФ РФ, ОМС, ОСС. Это контракт наследуется от контракта Owner, так как изменение кошельков должно быть доступно только создателю контракта.

В листинге 2 описан код контракта.

Листинг 2

```
pragma solidity 0.8.4;

import { Owner } from «./owner.sol»;

contract TaxInstitutionWallets is Owner {
    address private NdflWallet;
    address private PensionWallet;
    address private OmsWallet;
    address private OssWallet;

    function setNdflWallet(address wallet) public isOwner {
        NdflWallet = wallet;
    }

    function setPensionWallet(address wallet) public isOwner {
        PensionWallet = wallet;
    }

    function setOmsWallet(address wallet) public isOwner {
        OmsWallet = wallet;
    }

    function setOssWallett(address wallet) public isOwner {
        OssWallet = wallet;
    }

    function getNdflWallet() public isOwner view returns(address) {
        return NdflWallet;
```

```
    }

    function getPensionWallet() public isOwner view returns(address) {
        return PensionWallet;
    }

    function getOmsWallet() public isOwner view returns(address) {
        return OmsWallet;
    }

    function getOssWallet() public isOwner view returns(address) {
        return OssWallet;
    }
}
```

В контракте есть 4 переменные для кошельков, а также функции для установки и просмотра значений в переменных. У каждой функции стоит модификатор isOwner, который даёт права вызова этой функции только владельцу контракта.

3.3. Контракт CalculatorPercent

Ещё одним вспомогательным компонентом будет контракт CalculatorPercent.

На данный момент в языке Solidity (язык для написания смарт-контрактов в сети Ethereum) отсутствуют числа с плавающий запятой, поэтому требуется переводить рубли в копейки. Этот контракт как раз и занимается переводом из рублей в копейки, а также считает проценты от переданной суммы. Его код представлен в листинге 3.

Листинг 3

```
pragma solidity 0.8.4;

contract CalculatorPercent {
    function toPenny(uint rubles) public pure returns(uint) {
        return rubles * 100;
    }

    function toRubles(uint penny) public pure returns(uint) {
        return penny / 100;
    }

    function toRublesAndPenny(uint penny) public pure returns(uint, uint) {
        uint resRubles = 0;
        uint resPenny = 0;
        if (penny >= 100) {
            resRubles = penny / 100;
        }

        resPenny = penny - resRubles * 100;
        return (resRubles, resPenny);
    }

    function getPercentOfAmount(uint rubles, uint8 penny, uint16 percent) public pure
returns(uint) {
        return (toPenny(rubles) + penny) / 100 * percent / 10;
    }
}
```

3.3. Контракт NalogSender

Самый главный контракт – NalogSender. Он предоставляет интерфейс работодателю для пополнения своего баланса, выплаты зарплаты сотруднику. В листинге 4 представлен его код.

Листинг 4

```
pragma solidity 0.8.4;

import { CalculatorPercent } from «./calculatorPercent.sol»;
import { TaxInstitutionWallets } from «./taxInstitutionWallets.sol»;

contract NalogSender is TaxInstitutionWallets {
    uint8 private ndflTax = 130;
    uint8 private pensionTax = 220;
    uint16 private omsTax = 51;
    uint16 private ossTax = 29;

    mapping(address => uint) taxpayers;

    CalculatorPercent calc = new CalculatorPercent();

    constructor() {}

    function getAmoutTax(uint rubles, uint8 penny) private returns(uint) {
        uint ndfl = calc.getPercentOfAmount(rubles, penny, ndflTax);
        uint pension = calc.getPercentOfAmount(rubles, penny, pensionTax);
        uint oms = calc.getPercentOfAmount(rubles, penny, omsTax);
        uint oss = calc.getPercentOfAmount(rubles, penny, ossTax);
        return ndfl + pension + oms + oss;
    }

     function sendSalary(uint rubles, uint8 penny, address payable wallet) public
returns(uint) {
        uint amoutTax = getAmoutTax(rubles, penny);
        uint salary = calc.toPenny(rubles) + penny;
        uint amout = amoutTax + salary;
        require(taxpayers[msg.sender] >= amout, «Not enough balance»);
        sendTaxes(rubles, penny);
        wallet.transfer(salary);
        taxpayers[msg.sender] -= salary;
        return salary + amoutTax;
    }

    function sendTaxes(uint rubles, uint8 penny) private {
        uint ndfl = calc.getPercentOfAmount(rubles, penny, ndflTax);
        payable(this.getNdflWallet()).transfer(ndfl);
        taxpayers[msg.sender] -= ndfl;
        uint pension = calc.getPercentOfAmount(rubles, penny, pensionTax);
        payable(this.getPensionWallet()).transfer(pension);
        taxpayers[msg.sender] -= pension;
        uint oms = calc.getPercentOfAmount(rubles, penny, omsTax);
        payable(this.getOmsWallet()).transfer(oms);
        taxpayers[msg.sender] -= oms;
        uint oss = calc.getPercentOfAmount(rubles, penny, ossTax);
        payable(this.getOssWallet()).transfer(oss);
        taxpayers[msg.sender] -= oss;
    }
```

```
    function deposit() public payable {
        taxpayers[msg.sender] += msg.value;
    }

    function myBalance() public view returns(uint) {
        return taxpayers[msg.sender];
    }

    function withdraw() public {
        uint amount = taxpayers[msg.sender];
        taxpayers[msg.sender] = 0;
        payable(msg.sender).transfer(amount);
    }
}
```

В этом контракте есть 4 приватные переменные, в которых записан процент налога. Также есть переменная taxpayers, которая хранит в себе балансы всех кошельков, которые переводили на баланс контракта какие-то средства. Дело в том, что контракт может переводить средства только из своего баланса. Этот баланс общий на весь контракт. Поэтому нужно создать переменную, в который будет хранится информация о балансе каждого работодателя на этом контракте. Функция sendSalary служит для выплаты зарплаты сотруднику. В неё передаются количество рублей, количество копеек и адрес кошелька сотрудника. Далее внутри неё высчитывается налог, переводится на кошельки из контракта TaxInstitutionWallets и потом отправляются сотруднику.

Если при выполнении контракта, случится ошибка все средства будут возвращены на баланс работодателя.

Заключение

В работе рассмотрены возможности применения технологии Blockchain для улучшения налоговой системы. Были разработаны смарт-контракты, упрощающие уплату налогов для работодателя. Внедрение блокчейна в бизнес-процессы налоговой системы позволит сделать эти процессы быстрее, проще, прозрачнее и дешевле.

Литература

1. World Economic Forum Survey Projects Blockchain ‘Tipping Point’ by 2023. – Режим доступа: https://www.coindesk.com/world-economic-forum-governments-blockchain (дата обращения: 14.04.2021).

2. Что такое блокчейн и зачем он нужен. – Режим доступа: https://habr.com/ru/company/bitfury/blog/321474/ (дата обращения: 14.04.2021).

3. Что такое смарт-контракты? – Режим доступа: https://habr.com/ru/post/448056/ (дата обращения: 14.04.2021).

4. Сколько налогов платит работодатель. – Режим доступа: https://kontur.ru/articles/4845 (дата обращения: 14.04.2021).

Сапронов Александр Юрьевич – студент 4-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: scrib2@yandex.ru

Авсеева Ольга Владимировна (научный руководитель) – канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: olga-avseeva@mail.ru

УДК 004.896:621.865

АППАРАТНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ПРОХОЖДЕНИЯ ЛАБИРИНТА ПО ПРАВИЛУ ЛЕВОЙ РУКИ

Д. Ю. Власов, А. О. Северов

Воронежский государственный университет

Введение

Лабиринт – структура (чаще всего в двухмерном пространстве), состоящая из запутанных путей к выходу и путей, ведущих в тупик. Главный смысл лабиринта – это найти правильный путь к выходу, среди всех ответвлений и тупиков. Люди решали задачу прохождения лабиринта еще с древних времён. Цели прохождения лабиринта были разные, начиная от пересечения местности заканчивая религиозными обрядами. Первые попытки автоматизировать поиск выхода из лабиринта были предприняты в середине 20 века. В 1950 году учёный Клод Шеннон создал механическую мышь «Тесей», которая могла находит и запоминать путь в лабиринте путём переключения электромеханических реле под полом лабиринта. Это был один из первых экспериментов в области искусственного интеллекта. С развитием информатики стали появляться различные методы, которые можно использовать в задаче прохождения лабиринта. Таковыми являются: алгоритмы, backtracking, машинное обучение.

В данной статье рассматривается реализация алгоритма прохождения лабиринта по правилу «одной руки» на примере программного комплекса, состоящего из робота, являющимся искусственным агентом и мобильным приложением, осуществляющего функции пульта управления. Робот способен проходить односвязный лабиринт, а приложение имеет возможность с ним взаимодействовать.

1. Алгоритм прохождения лабиринта

1.1. Правило левой руки

Правило «одной руки» является одним из самых простых правил, позволяющих пройти односвязный лабиринт. Вероятно, этот алгоритм был известен еще древним грекам, несмотря на то, что придётся зайти во множество тупиков, конечная цель будет достигнута. Правило левой руки работает следующим образом:

1. Объект движется вперёд, пока не наткнётся на поворот, развилку, тупик или финиш.
2. Если поворот или развилка, то объект поворачивает в следующем приоритете: налево, прямо, направо и переходит к пункту 1.
3. Если тупик, то объект разворачивается и переходит к пункту 1.
4. Если финиш, то объект заканчивает своё движение.

На рис. 1 изображена схема данного алгоритма.

1.2. Поиск оптимального пути

После прохождения лабиринта по правилу левой руки, робот отправляет данные о своём маршруте, записанным следующим образом: СЛРЛВПФ, где С – старт, Л – поворот налево, П – поворот направо, В – вперёд, Р – разворот, Ф – финиш.

Для нахождения оптимального маршрута были разработаны правила, позволяющие сократить запись пройденного маршрута. Сокращения были получены в результате присвоения

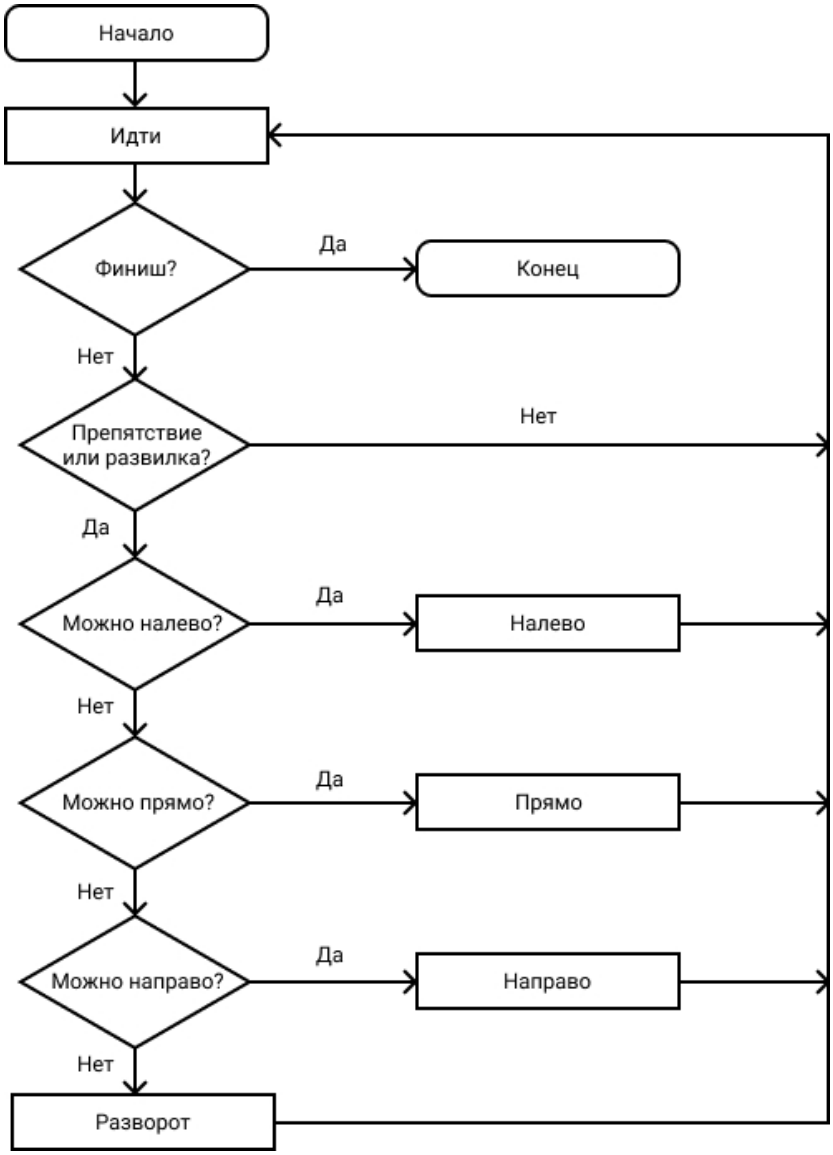


Рис. 1. Алгоритм правила «левой руки»

буквам их фактического значения в градусах: Л = –90, П = 90, В = 0, Р = 180. Путём сложения следующих градусов была получен следующий список сокращений:

- ЛРЛ = В (–90+180–90=0);
- ЛРП = Р (–90+180+90=180);
- ЛРВ = П (–90+180+0=90);
- ПРЛ = Р (90+180–90=180);
- ВРЛ = П (0+180–90=90).

Иные комбинации букв при использовании данного алгоритма невозможны. После каждого сокращения происходит перезапись всего измененного пути. Алгоритм работает до тех пор, пока встречаются выше перечисленные комбинации.

1.3. Пример использования правила «левой руки» и оптимизации

На рис. 2 представлен односвязный лабиринт. После прохождения этого лабиринта по правилу левой руки получается следующий маршрут: СЛРЛЛЛРЛРЛРЛЛППФ. Данный маршрут требует оптимизации, который происходит следующим образом:

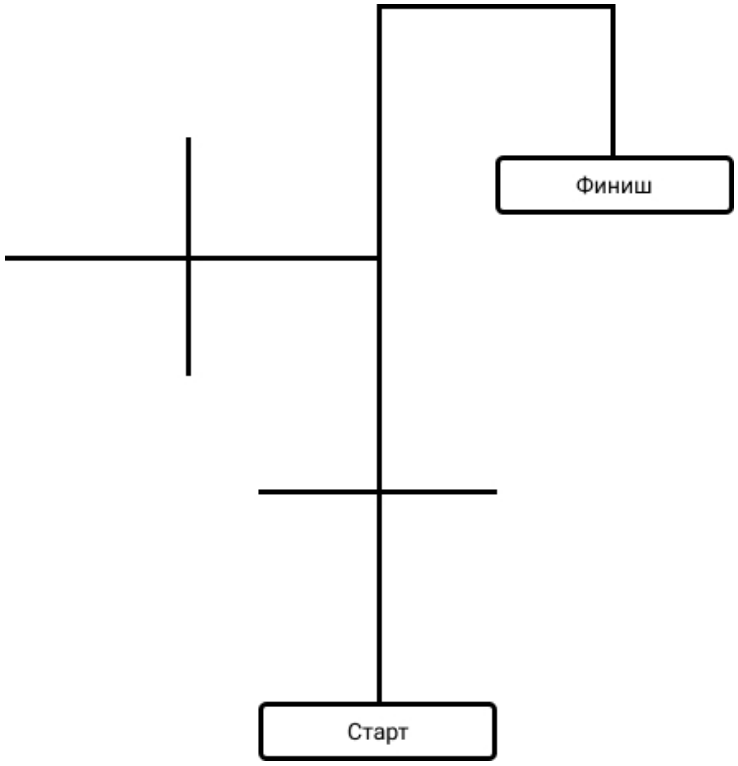


Рис. 2. Односвязный лабиринт

1. С(**ЛРЛ**ЛЛРЛРЛРЛЛППП)Ф.
2. С(ВЛ**ЛРЛ**РЛРЛЛППП)Ф.
3. С(ВЛ**ВРЛ**РЛЛППП)Ф.
4. С(ВЛ**ПРЛ**ЛППП)Ф.
5. С(В**ЛРЛ**ПП)Ф.
6. С(ВВПП)Ф.

Оптимальный путь: СВВППФ.

2. Реализация

2.1. Структура программного комплекса

Программный комплекс осуществляет прохождение односвязного лабиринта и поиск оптимального пути. Он состоит из двух частей: робота (искусственного агента) и приложения на Android. Эти две составляющие осуществляют обмен сообщениями при помощи технологии Bluetooth.

2.2. Устройство робота

Робот состоит из следующих компонентов: плата Arduino UNO, два мотора-редуктора, два колеса, шаровое колесо, датчик цвета, аккумулятор, Bluetooth модуль, деревянная основа, провода. На рис. 3 представлена схема подключения компонентов робота.

Робот может выполнять следующие функции:

- проходить новый лабиринт по правилу «левой руки»;
- обмениваться сообщениями с мобильным приложением;
- проходить известный лабиринт по полученному маршруту.

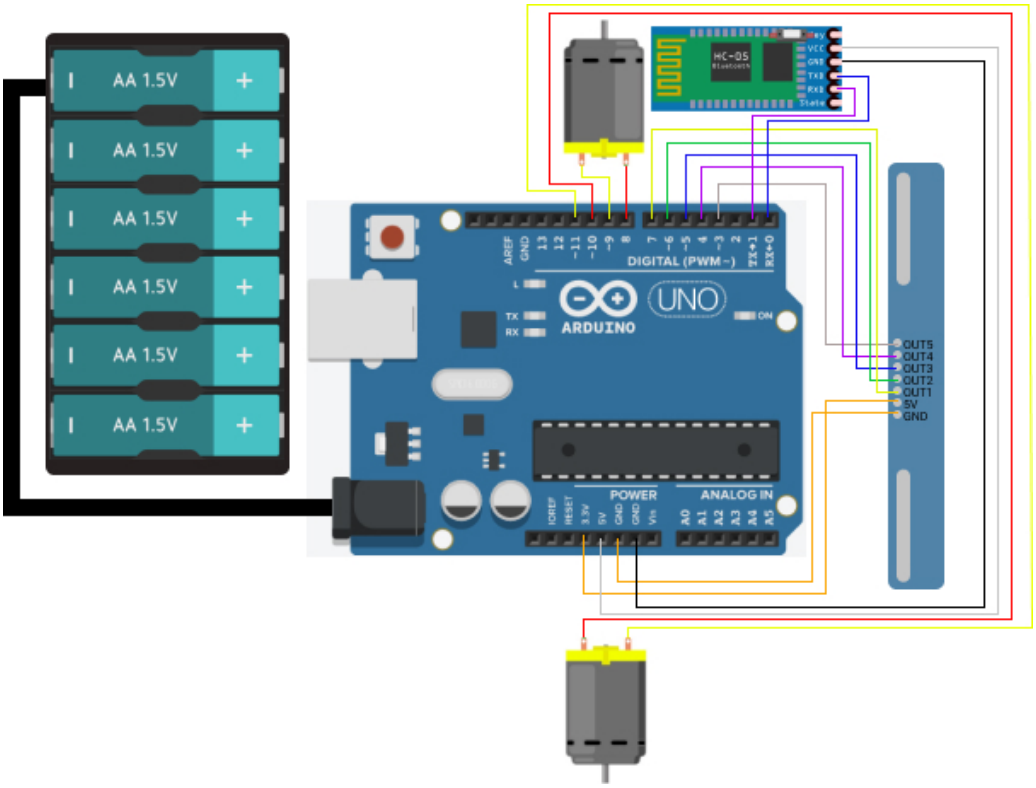


Рис. 3. Схема подключения компонентов робота

2.3. Логика мобильного приложения

Мобильное приложение имеет 3 кнопки:

1. Запустить – активирует робота для прохождения маршрута.
2. Построить маршрут – строит кратчайший маршрут по алгоритму поиска оптимального пути и отправляет его роботу.
3. Остановить – прекращает работу робота.

После запуска приложения активна лишь кнопка «Запустить». При движении робота кнопка «Запустить» становится неактивной, а кнопка «Остановить» активной. После приёма пройденного маршрута имеется возможность «Построить маршрут».

3. Усовершенствование алгоритма

На данный момент комплекс может выполнить следующие действия: пройти односвязный лабиринт и построить оптимальный маршрут. Данной функциональности хватает для решения узкого класса задач. Возможности программного комплекса можно усовершенствовать путём доработки логической и аппаратной части.

Усовершенствование аппаратной части:

1. Установка датчика, определяющего пройденное расстояние.
2. Установка механизма, позволяющего оставлять метки на лабиринте.

Эти усовершенствования необходимы для дальнейшего улучшения логической части комплекса.

Усовершенствование логической части:

1. Изменение алгоритма для возможности прохождения двусвязных лабиринтов.
2. Поиск всех возможных решений лабиринта.
3. Выбор оптимального маршрута с учётом расстояния при наличии нескольких путей к финишу.

Данные доработки позволят решать комплексу более сложные задачи, включая задачи практического применения.

Заключение

В результате проделанной работы был создан программный комплекс, который может проходить односвязный лабиринт по правилу «левой руки» и строить оптимальный маршрут. Данный комплекс имеет возможность усовершенствования для решения более сложных учебных и практических задач.

Литература

1. *Филлипс, Б.* Android. Программирование для профессионалов. 3 изд. / Филлипс Б., Стюарт К., Марсикано К. – Санкт-Петербург : Питер, 2017. – 688 с.

2. *Васильев, А. Е.* Микроконтроллеры. Разработка встраиваемых приложений / А. Е. Васильев. – СПб. : BHV, 2012. – 304 с.

3. *Соммер, У.* Программирование микроконтроллерных плат Arduino / Freeduino. – СПб. : БХВПетербург, 2012. – 256 с.

4. *Трахтенброт, Б. А.* Алгоритмы и машинное решение задач / Б. А. Трахтенброт – Москва : ГОСТЕХИЗДАТ, 1957. – 96 с.

Власов Денис Юрьевич – магистрант 1-го года обучения кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: den10pmm@mail.ru

Северов Артём Олегович – магистрант 1-го года обучения кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: fisaforyou@mail.ru

Воронина Ирина Евгеньевна (научный руководитель) – д-р техн. наук, доц., профессор кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: irina.voronina@gmail.com

УДК 004.056.55

ХАРАКТЕРИСТИКА И СРАВНЕНИЕ ЦИФРОВЫХ ПОДПИСЕЙ DSS, RSA, ГОСТ 34.10-2018

П. Д. Семиколенов, В. Г. Ляликова

Воронежский государственный университет

Введение

В данной статье рассматриваются такие криптографических алгоритмы, как DSA, RSA, Эль-Гамаль с использованием эллиптических кривых, для наглядного разбора и визуального сравнения их характеристик и основ, а также составляющих цифровых подписей на основе данных алгоритмов, DSS, RSA, ГОСТ 34.10-2018 соответственно.

1. DSA(DSS)

DSA или Digital Signature Algorithm – криптографический алгоритм с использованием закрытого ключа для создания электронной подписи, но не для шифрования. В цифровой подписи на основе данного, как и всех последующих разбираемых в данной статье, алгоритма используется метод асимметричного шифрования, в котором существуют два ключа – закрытый, для создания цифровой подписи, и открытый, для проверки ее подлинности.

Алгоритм основан на трудности вычисления дискретных логарифмов в конечном поле, реализующий частный вариант криптографической схемы Эль-Гамаля. Чтобы лучше понять строение алгоритма, разберем его более подробно на примере стандарта цифровой подписи DSS.

Размер ключа в DSS составляет 512–1024 бит, а размер подписи 160 бит.

Для алгоритмов на основе схемы Эль-Гамаля цифровая подпись фактически создается не из исходного сообщения, а из Дайджест сообщения или по-другому, хеша сообщения, результата обработки исходного текста хеш-функцией.

1.1. Хеш функция

Изначально в стандарте DSS использовалась хеш-функция SHA-1, что на выходе давал хеш длинной 160 бит, но в данный момент используются криптографический алгоритм из семейства SHA-2, представителями которого являются SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224 (числа говорят о размере выходного хеша).

Хеш-функции семейства SHA-2 построены на основе структуры Меркла – Дамгора метод построения криптографических хеш-функций, предусматривающий разбиение входных сообщений произвольной длины на блоки фиксированной длины и работающий с ними по очереди с помощью функции сжатия, каждый раз принимая входной блок с выходным от предыдущего прохода. Исходное сообщение после дополнения разбивается на блоки, каждый блок на 16 слов. Алгоритм пропускает каждый блок сообщения через цикл с 64 или 80 итерациями (раундами). На каждой итерации 2 слова преобразуются, функцию преобразования задают остальные слова. Результаты обработки каждого блока складываются, сумма является значением хеш-функции. Тем не менее, инициализация внутреннего состояния производится результатом обработки предыдущего блока. Поэтому независимо обрабатывать блоки и складывать результаты нельзя.

1.2. Алгоритм

В алгоритме используются следующие параметры:

1. Выбор простого числа p, размерность которого N в битах совпадает с размерностью в битах значений хеш-функции $H(x)$.

2. Выбор простого числа p, такого, что $(p-1)$ делится на q. Битовая длина p обозначается $L(2^{L-1} < p < 2^L)$.

3. Выбор числа g такого, что его мультипликативный порядок по модулю p равен q. Для его вычисления можно воспользоваться формулой $g = h^{(p-1)/q} \bmod p$, где h – некоторое произвольное число, $h \in (1; p-1)$ такое, что $g \neq 1$.

Подпись сообщения

1. Выбор случайного числа $k \in (0, q)$
2. Вычисление $r = (g^k \bmod p) \bmod q$
3. Выбор другого k, если $r = 0$
4. Вычисление $s = k^{-1}(H(m) + x \times r) \bmod q$
5. Выбор другого k, если $s = 0$
6. Подписью является пара (r, s) общей длины $2N$

Проверка подписи

1. Вычисление $w = s^{-1} \bmod q$
2. Вычисление $u_1 = h(m) \times w \bmod q$
3. Вычисление $u_2 = r \times w \bmod q$
4. Вычисление $u = (q^{u_1} \times y^{y_2} \bmod p) \bmod q$
5. Подпись верна, если $u = r$

2. RSA

RSA или Rivest, Shamir и Adleman – криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи.

Факторизацией натурального числа называется его разложение в произведение простых множителей. Существование и единственность такого разложения следует из основной теоремы арифметики. В отличие от задачи распознавания простоты числа, факторизация предположительно является вычислительно сложной задачей. В настоящее время неизвестно, существует ли эффективный не квантовый алгоритм факторизации целых чисел.

В данный момент система RSA может считаться надежной, если длина ее ключей составляет 1024-2048 и более бит.

2.1. Алгоритм

В алгоритме используются следующие параметры:

1. Выбираются два различных случайных простых p и q числа заданного размера
2. Вычисляется их произведение $n = p \times q$, которое называется модулем
3. Вычисляется значение функции Эйлера от числа: $\varphi(n) = (p-1) \times (q-1)$
4. Выбирается целое число $e(1 < e < \varphi(n))$, взаимно простое со значением функции $\varphi(n)$
5. Вычисляется число d, мультипликативнообратное к числу e по модулю $\varphi(n)$:

$$d \times e \equiv 1 \pmod{\varphi(n)}$$

6. пара (e, n) публикуется в качестве открытого ключа RSA
7. пара (d, n) играет роль закрытого ключа RSA и держится в секрете

Подпись сообщения

1. Взять открытый текст m
2. Создать цифровую подпись s с помощью своего секретного ключа (d, n):

$$s = S_A(m) = m^d \mod n$$

3. Передать пару (m, s), состоящую из сообщения и подписи

Проверка подписи

1. Принять пару (m, s)
2. Взять открытый ключ (e, n)
3. Вычислить прообраз сообщения из подписи $m' = P_A(s) = s^e \mod n$
4. Сравнить m' и m. Если они равны, то подпись верна

3. Эль-Гамаль для эллиптических кривых(ГОСТ 34.10-2018)

Рассмотрим реализацию данного алгоритма более подробнее на Российском стандарте цифровой подписи ГОСТ 34.10-2018.

Размерность ключей 256-1024 бит.

Стойкость данного алгоритма все также опирается на сложность вычисления дискретного логарифма, но в отличие от DSA, дискретный логарифм вычисляется в группе точек эллиптической кривой, а также стойкости используемой в стандарте хеш-функции.

3.1. Хеш-функция

«Стрибог» – криптографический алгоритм вычисления хеш-функции с размером блока входных данных 512 бит и размером хеш-кода 256 или 512 бит.

В хеш-функции важным элементом является функция сжатия, использующая конструкции Миагути – Пренеля: h, m – вектора, поступающие на вход функции сжатия; $g(h, m)$ – результат функции сжатия; E – блочный шифр с длиной блока и ключа 512 бит. В качестве блочного шифра в хеш-функции ГОСТ Р 34.11-2012 взят XSPL-шифр. Этот шифр состоит из следующих преобразований:

- сложение по модулю 2;
- преобразование замены или подстановки. Обозначается S-преобразование;
- преобразование перестановки. Обозначается P-преобразование;
- линейное преобразование. Обозначается L-преобразование.

В основу хеш-функции положена итерационная конструкция Меркла – Дамгора с использованием MD-усиления. Под MD-усилением понимается дополнение неполного блока при вычислении хеш-функции до полного путём добавления вектора (0 … 01) такой длины, чтобы получился полный блок. Из дополнительных элементов нужно отметить следующие:

- завершающее преобразование, которое заключается в том, что функция сжатия применяется к контрольной сумме всех блоков сообщения по модулю 2512;
- при вычислении хеш-кода на каждой итерации применяются разные функции сжатия. Можно сказать, что функция сжатия зависит от номера итерации.

Кратко описание хеш-функции ГОСТ Р 34.11-2012 можно представить следующим образом. На вход хеш-функции подается сообщение произвольного размера. Далее сообщение разбивается на блоки по 512 бит, если размер сообщения не кратен 512, то оно дополняется необходимым количеством бит. Потом итерационно используется функция сжатия, в результате действия которой обновляется внутреннее состояние хеш-функции. Также вычисляется контрольная сумма блоков и число обработанных бит. Когда обработаны все блоки исходного сообщения, производятся ещё два вычисления, которые завершают вычисление хеш-функции:

• обработка функцией сжатия блока с общей длиной сообщения.
• обработка функцией сжатия блока с контрольной суммой.

3.2. Алгоритм цифровой подписи

В алгоритме используются следующие параметры:
• p – простое число, модуль эллиптической кривой
• a, b – коэффициенты эллиптической кривой
• m – целое число, порядок подгруппы кривой, отлично от p
• q – простое число, порядок некоторой циклической подгруппы, $m = n \times q$, для некоторого $n \in N$, при этом $2254 < q < 2256$.
• $P = (x_p, y_p)$ – базовая точка подгруппы порядка q
• $h(M)$ – хеш-функция от сообщения M
• d – целое число, ключ шифрования, $0 < d < q$
• $Q = (x_Q, y_Q)$ – ключ расшифрования, $Q = d \times P$

Подпись сообщения

1. Вычисление хеш-функции от сообщения $M : \overline{h} = h(M)$
2. Вычисление $e = z \bmod q$, и если $e = 0$, положить $e = 1$. Где z – целое число, соответствующее $\overline{h}$.
3. Генерация случайного числа k такого, что $0 < k < q$.
4. Вычисление точки эллиптической кривой $C = k \times P$, и по ней нахождение $r = x_c \bmod q$, где x_c – это координата x точки C. Если $r = 0$, возвращаемся к предыдущему шагу.
5. Нахождение $s = (r \times d + k \times e) \bmod q$. Если $s = 0$, возвращаемся к шагу 3.
6. Формирование цифровой подписи $\xi = (\overline{r} \mid \overline{s})$, где $\overline{r}$ и $\overline{s}$ – векторы, соответствующие r и s.

Проверка подписи

1. Вычисление по цифровой подписи ξ чисел r и s, учитывая, что $\xi = (\overline{r} \mid \overline{s})$, где r и s – числа, соответствующие векторам $\overline{r}$ и $\overline{s}$. Если хотя бы одно из неравенств $r < q$ и $s < q$ неверно, то подпись неправильная.
2. Вычисление хеш-функции от сообщения $M : \overline{h} = h(M)$.
3. Вычисление $e = z \bmod q$, и если $e = 0$, положить $e = 1$. Где z – целое число соответствующее $\overline{h}$.
4. Вычисление $v = e^{-1} \bmod q$.
5. Вычисление $z_1 = s \times v \bmod q$ и $z_2 = -r \times v \bmod q$.
6. Вычисление точки эллиптической кривой $C = z_1 \times P + z_2 \times Q$. И определение $R = x_c \bmod q$, где x_c – координата x точки C.
7. В случае равенства $R = r$ подпись правильная, иначе – неправильная.

Заключение

В данной статье были разобраны характеристики параметры таких алгоритмов как: DSA, RSA, Эль-Гамаль на эллиптических кривых, а также реализующие данные алгоритмы стандарты цифровых подписей. Были наглядно продемонстрированы их подходы к шифрованию, выбору хеш-функций и методов формирования цифровых подписей, а также их проверки. Если сжать информацию, написанную в данной статье, то получим следующие (цифровая подпись: длина ключей, хеш-функция, алгоритм цифровой подписи):
• DSS: 512–1024 бит, SHA-2(структура Меркла – Дамгора), DSA(Эль-Гамаль).
• RSA: 512-2048 бит, - , RSA(факторизация).
• ГОСТ 34.10-2018: 256-512, Стрибог(функция сжатия – Миагути – Пренеля, основа конструкция Меркла — Дамгора), схема Эль-гамаля на эллиптических кривых.

Цифровая подпись, основанная на схеме Эль-Гамаля на эллиптических кривых, имеет значительно меньшую длину ключей, что является существенным преимуществом перед другими алгоритмами, а в частности DSS, что также использует данный алгоритм(Эль-Гамаль). Связано это с тем, что схема строится над полями эллиптических кривых, а не действительных чисел.

Литература

1. *Shawe-Taylor J.* Generating strong primes / J. Shawe-Taylor // Electronics Letters – 31 July 1986. – Vol, 22, Iss. 16. – P. 875–877.

2. Digital Signature Standard. – Режим доступа: https://ru.wikipedia.org/wiki/Digital_Signature_Standard (дата обращения 24.04.2021).

3. Алгоритм DSA. – Режим доступа: http://wincrypt.chat.ru/dsa.htm (дата обращения 24.04.2021).

4. *Diffie W., Hellman M. E.* New Directions in Cryptography // IEEE Trans. Inf. Theory / F. Kschischang — IEEE, 1976. — Vol. 22, Iss. 6. — P. 644–654.

5. Elliptic Curve Cryptography: a gentle introduction – Режим доступа: https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/ – (Дата обращения 24.04.2021).

6. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи – Введ. 2019-06-01 – Москва Стандартиформ 2018.

Семиколенов Павел Дмитриевич – магистрант 2-го года обучения кафедры ERP-систем и бизнес процессов факультета ПММ Воронежского государственного университета. E-mail: pavelsemikolenov.y@yandex.ru

Ляликова Виктория Геннадиевна (научный руководитель) – канд. физ.-мат. наук, доцент кафедры ERP-систем и бизнес процессов факультета ПММ Воронежского государственного университета. E-mail: vikalg@yandex.ru

УДК 515.14, 004.8

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПАКЕТОВ ДЛЯ ВЫЧИСЛЕНИЯ УСТОЙЧИВЫХ ГОМОЛОГИЙ

П. М. Снопов

Воронежский государственный университет

Введение

В анализе данных существует так называемая гипотеза о многообразии, которая утверждает, что реальные данные, представленные облаком точек из $\mathbb{R}^n$ лежат на некотором d-многообразии, где $d \le n$. Персистентные гомологии позволяют узнать топологические характеристики этого многообразия(например, узнать его группы гомологий). Они являются основным инструментом топологического анализа данных – относительного новой области анализа данных, которая, опираясь на современные методы чистой математики, позволяет более изучать топологические свойства объектов, которые описываются облаком точек [3, 4].

Целью данной работы является сравнительный анализ пакетов для вычисления устойчивых гомологий с интерфейсом на языке Python, а также сравнение скорости работы данных пакетов на синтетических данных.

1. Некоторые теоретические сведения [1, 3, 4, 8]

Основными объектами в ТАД являются симплициальные комплексы.

Определение 1. *Симплициальный комплекс* K *– это множество симплексов, т. е. выпуклых оболочек набора* $n+1$ *точек* $\in \mathbb{R}^p$, *таких, что векторы* $x_1 - x_0, \ldots, x_n - x_0$ *линейно независимы, при этом*

- *для каждого симплекса из* K *его грани тоже лежат в* K,
- *пересечение любых двух симплексов* $\sigma, \tau \in K$ *либо пусто, либо является гранью и* σ, *и* τ.

Имея облако точек X, существует много способов построения симплициальных комплексов по нему. Наиболее популярные – симплициальные комплексы Вьеториса – Рипса и Чеха (рис. 1).

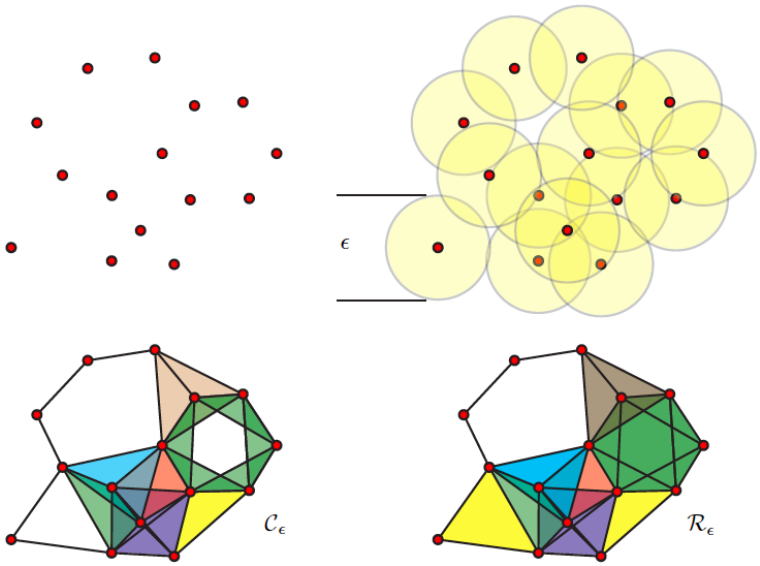


Рис. 1. Комплексы Чеха и Вьеториса – Рипса

Имея симплициальный комплекс, можно посчитать (симплициальные) гомологии: Пусть K – симплициальный комплекс и $k \geq 0$. Группой k-цепей $C_k(K)$ симплициального комплекса K называют (свободную абелеву) группу, элементами которого являются формальные суммы k-симплексов K, то есть

$$c = \sum_i \varepsilon_i \sigma_i, \ \ \varepsilon_i \in \mathbb{Z},$$

где конечное число $\varepsilon_i \neq 0$, σ_i – k-симплекс. Границей k-цепи $\sum_i \varepsilon_i \sigma_i$ называют $(k-1)$-цепь

$$\partial_k(\sigma) = \sum_{j=0}^{k} (-1)^j \sum_i \varepsilon_i \partial_j \sigma_i,$$

где $\partial_j \sigma_i = \partial_j [v_0, ..., v_k] = [v_0, ..., \hat{v}_j, ..., v_k]$ – $(k-1)$-симплекс, порожденный всеми вершинами, кроме вершины v_j. Гомоморфизм $\partial_k : C_k(K) \to C_{k-1}(K)$ называют граничным оператором. Он удовлетворяет следующему свойству:

$$\partial_{k-1} \circ \partial_k = 0.$$

То есть $\operatorname{im} \partial_{k+1} \leq \ker \partial_k \leq C_k(K)$. Последовательность $C_k(K)$ и ∂_k называется цепным комплексом

$$\cdots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \cdots \xrightarrow{\partial_1} C_0.$$

Определение 2. *k-й группой гомологий симплициального комплекса K называют следующее фактор-пространство:*

$$H_k(K) = \ker \partial_k \Big/ \operatorname{im} \partial_{k+1}.$$

Тогда k-е число Бетти – размерность k-й группы гомологий: $\beta_k(K) = \dim H_k(K)$.

При $k = 0$ число Бетти описывает количество компонент связности данного пространства. При $k = 1$ – количество циклов. При $k = 2$ число Бетти описывает количество «полостей».

Таблица 1

Первые числа Бетти для некоторых пространств

Пространство	β_0	β_1	β_2
Pt	1	0	0
D^2	1	0	0
Треугольник	1	0	0
Граница треугольника	1	1	0
S^1	1	1	0
S^2	1	0	1
$\mathbb{T}^2 = S^1 \times S^1$	1	2	1

Фильтрацией симплициального комплекса K называют вложенное семейство подкомплексов $(K_\tau)_{\tau \in T}$, где $T \subseteq \mathbb{R}$, такое, что если $\tau < \tau'$, то $K_\tau \subseteq K_{\tau'}$. Имея фильтрацию $(K_\tau)_{\tau \in T}$, можно отслеживать изменение $H_k(K_\tau)$ с ростом τ: могут появляться новые компоненты связности, уже существующие могут объединяться в одну компоненту, могут появляться циклы и «пустоты», соответствующие 1 и 2 группе гомологий.

Определение 3. *k-ми устойчивыми гомологиями фильтрованного комплекса $(K_\tau)_{\tau \in T}$ называют проиндексированное семейство абелевых групп $H_n(T) = \{(H_n(K_\tau))_{\tau \in T}$ вместе с семейством гомомомрфизмов $(H_n(K_\tau) \to H_n(K_{\tau'}))_{\tau \leq \tau'}\}$.*

Такое семейство абелевых групп вместе с морфизмами называют персистентным модулем. Основная теорема про персистентные модули – интервальное разложение персистентных модулей. Для этого обозначим за $\mathbb{V}(I)$ персистентный модуль, имеющий вид

$$\mathbb{V}(I)_t = \begin{cases} F, & t \in I, \\ 0, & \text{иначе}, \end{cases}$$

где F – поле, над которым рассматриваются векторные пространства персистентного модуля. Такие персистентные модули называются интервальными.

Теорема (об интервальном разложении). *Конечнопорожденный персистентный модуль* $\mathbb{V}$ *раскладывается в прямую сумму некоторого семейства интервальных персистентных модулей, причем это разложение единственно с точностью до перестановки:*

$$\mathbb{V} \cong \bigoplus_{s \in S} \mathbb{V}(I_s).$$

Применяя эту теорему к персистентным гомологиям, получаем, что информацию о группах гомологий фильтрации можно записывать полученную информацию в виде диаграммы, называемой баркодом (рис. 2), содержащей отрезки, которые отвечают за время жизни свойств, которые как раз и характеризуются гомологиями.

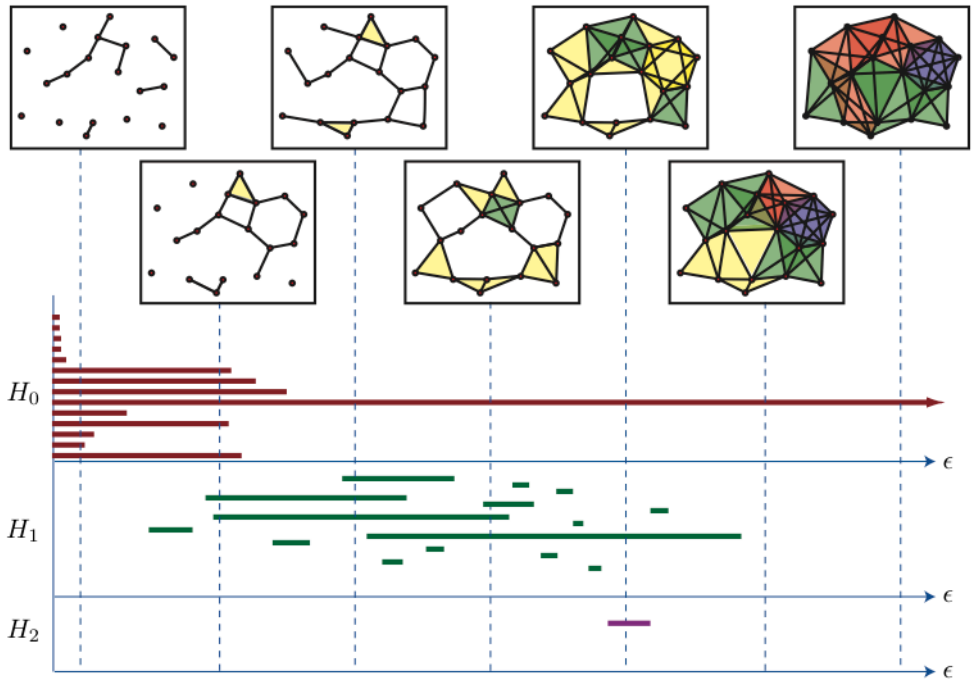


Рис. 2. Баркод

Из имеющегося баркода можно получить диаграмму персистентности (рис. 3). Она строится следующим образом: каждый интервал баркода имеет начало t_{birth} и конец t_{death}. На персистентной диаграмме каждый интервал баркода изображается в виде точки с координатами (t_{birth}, t_{death}). Чем дальше точка на персистентной диаграмме от диагонали, тем она важнее – эта точка сигнализирует о наличии «n-мерной дырки».

Теорема об интервальном разложении корректно определяет такие персистентные диаграммы.

2. Сравнительный анализ пакетов для вычисления устойчивых гомологий

В настоящее время существует несколько библиотек с интерфейсом на Python, позволяющих вычислять устойчивые гомологии: Dionysus [9], Giotto-TDA [6], GUDHI [7], Ripser [11] – часть более обширной библиотеки Scikit-TDA [12] для топологического анализа данных. Пакеты Dionysus и GUDHI существуют уже длительное время, и поэтому более надежны в работе, когда как Scikit-TDA и Giotto-TDA являются более новыми, но быстро развивающимися библиотеками.

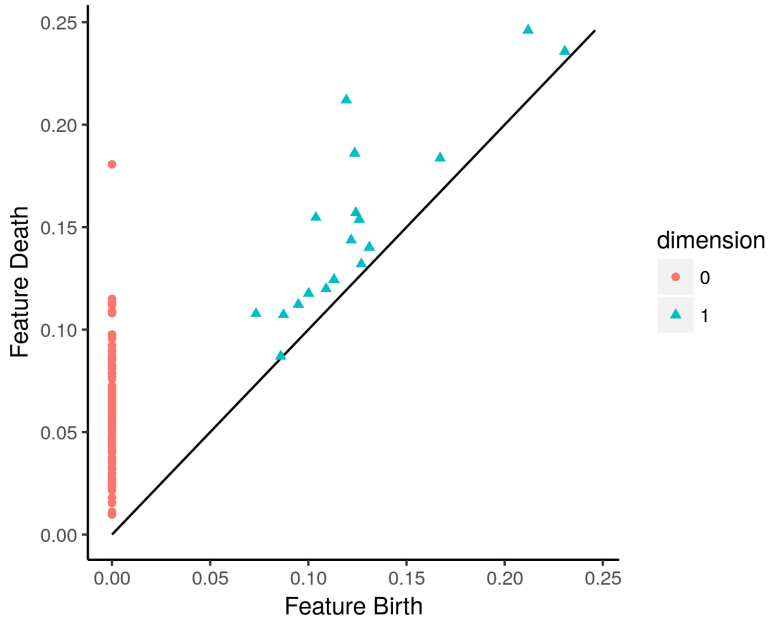


Рис. 3. Персистентная диаграмма

Сравнительный анализ этих пакетов представлен в табл. 2.

Следующие алгоритмы используются для вычисления устойчивых гомологий:

• Стандартный алгоритм [4], заключающийся в приведении граничной матрицы фильтрации к ступенчатому виду.

• Twist algorithm [14], заключающийся в оптимизации стандартного алгоритма путем уменьшения размерности граничной матрицы фильтрации.

• Dual algorithm, заключающийся в вычислении устойчивых когомологий, что вычислительно более эффективно.

• Multifield algorithm [15], заключающийся в использовании других полей коэффициентов для более эффективного вычисления.

• Zigzag algorithm [16], также основанный на персистентных когомологиях, которые вычисляются более эффективно.

Таблица 2

Сравнительный анализ возможностей пакетов

	Giotto-TDA	GUDHI	Ripser	Dionysus
Алгоритмы для вычисления устойчивых гомологий	standard, twist, dual	standard, dual, multifield	standard, twist, dual	standard, dual, zigzag
Коэффициенты	$\mathbb{F}_p$	$\mathbb{F}_p$	$\mathbb{F}_p$	$\mathbb{F}_p$ (dual); $\mathbb{F}_2$ (standard, zigzag)
Гомологии	Симплициальные, кубические	Симплициальные, кубические	Симплициальные	Симплициальные
Фильтрации	Виеторис – Рипс, Чех, α	α, Виеторис – Рипс, Чех, кубические, нерв, Witness, Делоне	Виеторис – Рипс	Виеторис – Рипс, α, Чех
Визуализация	Диаграммы устойчивости	Диаграммы устойчивости, баркоды	Диаграмма устойчивости	Диаграммы устойчивости, баркоды

Из таблицы видно, что библиотека GUDHI представляет максимально различные способы построения фильтраций, а также различные алгоритмы для вычисления устойчивых гомологий. Эта библиотека предоставляет также возможность вычислять кубические гомологии.

Проведем анализ пакетов на синтетических тестах. Для этого воспользуемся пакетов Tadasets [10], который, как и Ripser, является частью библиотеки Scikit-TDA [12], и предоставляет набор синтетических датасетов, полезных для топологического анализа данных. Рассматриваемые синтетические датасеты – это не зашумленные облака точек в $\mathbb{R}^{26}$; количество точек $n = 2000$. Данные описывают следующие многообразия:

- Top(расстояние от центра центра до центра внутренней окружности $c = 2$, радиус внутренней окружности $a = 1$).
- Швейцарский рулет(длина рулета $r = 4$).
- 2-Сфера(радиус $r = 3.14$).
- 3-Сфера(радиус $r = 3.14$).
- Букет 2 окружностей («знак бесконечности»).

Во всех пакетах использовался стандартный алгоритм вычисления персистентных гомологий. Целью данного сравнения было нахождение пакета, который быстрее других вычисляет устойчивые гомологии, и для которого потребуются минимальные настройки. Все вычисления производились в среде Google Colab [13]. На основе данных сравнений можно сделать вывод, что библиотека GUDHI предоставляет наиболее эффективный способ вычисления устойчивых гомологий. В свою очередь, библиотека Dionysus тратила очень много времени на построение фильтрации, поэтому в вычислительном эксперименте ее результатов нет.

Таблица 3

Тест скорости работы пакетов

Wall-time sec.					
	Top	Швейцарский рулет	2-Сфера	3-Сфера	«Знак бесконечности»
Ripser	26.3	392	6.84	13.5	50.4
Gudhi	2.2	2.21	2.27	2.27	1.51
Giotto-TDA	14	264	4.14	7.04	29.1
CPU time sec.					
	Top	Швейцарский рулет	2-Сфера	3-Сфера	«Знак бесконечности»
Ripser	22.4	380	6.2	11.7	50.4
Gudhi	2.12	2.15	2.21	2.23	1.45
Giotto-TDA	13.8	263	4.07	6.72	28.9

Таким образом, библиотека GUDHI представляет наибольшее разнообразие методов построения фильтраций, алгоритмов вычисления устойчивых гомологий, а также является эффективной с вычислительной точки зрения.

Заключение

Персистентные гомологии являются основным инструментом топологического анализа данных. Благодаря гипотезе о многообразии, свое применение персистентные гомологии находят в различных областях анализа данных. Эта гипотеза утверждает, что за любым облаком точек из $\mathbb{R}^d$ лежит некоторое многообразие. Персистентные гомологии предоставляют инструмент для изучения топологических характеристик данного многообразия.

В данной работе был проведен сравнительный анализ, а также вычислительный эксперимент пакетов на Python, позволяющих вычислять устойчивые гомологии.

Литература

1. *Хатчер, А.* Алгебраическая топология / А. Хатчер. – Москва : МЦНМО, 2011. – 688 с.

2. *Chazal, F.* An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists /Chazal F., Michel B. – URL : https://arxiv.org/pdf/1710.04019.pdf

3. *Edelsbrunner, H.* Computational Topology An Introduction / H. Edelsbrunner, J. Harer. – AMS : Providence, 2009. – 241 p.

4. *Zomorodian, A.* Computing persistent homology / A. Zomorodian, G. Carlsson // Discrete Comput. Geom. – 2005. – Vol. 33, No 2. – P. 249–274. – URL : https://geometry.stanford.edu/papers/zc-cph-05/zc-cph-05.pdf

5. *Otter, N.* A roadmap for the computation of persistent homology / N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, H. A Harrington // EPJ Data Sci. – 2017. – Vol. 6, No 17. – URL : https://epj-datascience.springeropen.com/articles/10.1140/epjds/s13688-017-0109-5

6. Giotto-tda – библиотека для топологического анализа данных на Python. – URL : https://github.com/giotto-ai/giotto-tda

7. GUDHI – библиотека для топологического анализа данных на C++ с интерфейсом для Python. – URL : https://gudhi.inria.fr/

8. Chazal F. The Structure and Stability of Persistence Modules / F. Chazal, V. de Silva, M. Glisse, S. Outdot. – URL : https://arxiv.org/pdf/1207.3674.pdf

9. Dionysus 2 – библиотека для вычислений устойчивых гомологий, написанная на C++ и имеющая интерфейс на Python. – URL : https://www.mrzv.org/software/dionysus/

10. Tadasets – библиотека, содержащая синтетические датасеты для топологического анализа данных. – URL : https://github.com/scikit-tda/tadasets

11. Интерфейс для Ripser на Python. – URL : https://ripser.scikit-tda.org/en/latest/

12. Scikit-TDA – библиотека для топологического анализа данных, содержащая Ripser и Tadasets. – URL : https://scikit-tda.org/

13. Google Colab – Сервис Google, предоставляющий возможность запускать код, написанный на Python, в браузере, обладающий интерфейсом Jupyter Notebook, специально созданный для задач машинного обучения, анализа данных, а также образования. – URL : https://colab.research.google.com/

14. *Chen, C.* Persistent Homology Computation with a Twist / C. Chen, M. Kerber // EuroCG – 2011. – URL : https://eurocg11.inf.ethz.ch/abstracts/22.pdf

15. *Boissonnat, J.* Computing Persistent Homology with Various Coefficient Fields in a Single Pass / J.Boissonnat, C. Maria. – URL : https://arxiv.org/abs/2001.02960.

16. *Maria, C.* Computing Persistent Cohomology / C. Maria, S. Oudot. – URL: https://arxiv.org/abs/1608.06039.

Снопов Павел Михайлович – бакалавр 4-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: snopoffp@gmail.com

Леденева Татьяна Михайловна (научный руководитель) – д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: ledeneva-tm@yandex.ru

УДК 004.657

СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ДОКУМЕНТООРИЕНТИРОВАННОЙ СУБД MONGODB И РСУБД POSTGRESQL

В. Г. Строгонов

Воронежский государственный университет

Введение

Необходимость сравнения производительности различных СУБД появилась при проектировании архитектуры платформы для организации логирования программных продуктов. В данной платформе должны предоставляться возможности быстрой записи и чтения в базу данных. В настоящее время двумя наиболее часто используемыми типами баз данных являются реляционные и NoSQL БД. Несмотря на то, что СУБД NoSQL появились относительно недавно, они стали достаточно популярным благодаря своей способности быстро обрабатывать несвязные и неструктурированные данные.

В рамках данной работы рассматривается сравнение производительности баз данных двух типов. Главной задачей было провести анализ производительности NoSQL-системы MongoDB на простых операциях, встроенных в РСУБД (таких как INSERT, SELECT), относительно реляционной системы управления данными PostgreSQL.

1. Подготовка данных

Для сравнения было решено добавлять 1 миллион записей в коллекцию MongoDB и PostgreSQL таблицу. Алгоритм генерации записей для обоих СУБД используется одинаковый. Для большей реалистичности тестирования был использован генератор случайных чисел. Тестовый объект имеет следующую структуру:

- Id – случайное целое число в диапазоне от 1000 до 1000000;
- Level – случайный уровень логирования из списка: ERROR, WARN, INFO, DEBUG, TRACE;
- Text – случайно сгенерированный текст длинной от 1000 до 5000 символов;
- ConsumedMemory – случайное дробное число от 100 до 10000.

2. Проведение замеров

Замеры проводились на компьютере со следующими характеристиками:

- CPU: Intel Xeon CPU E5-2650 v2 8 * 2.60GHz;
- RAM: 16 GB;
- OS: Windows 10.

Версии СУБД:

- MongoDB версии 4.4.3;
- PostgreSQL версии 12.2.

При проведении замеров использовался инструмент Java Microbenchmark Harness. Он помогает оценить фактическую производительность, принимая во внимание прогрев JVM и оптимизацию кода.

2.1. Замер времени вставки записей без использования индексов

Для того чтобы усреднить время вставки, миллион записей был разбит на 4 равных части по 250 тысяч записей. Время генерации включено в общее время вставки. Результаты замера приведены в табл. 1 и табл. 2.

Таблица 1

Замер времени вставки СУБД MongoDB

Операция	**Затраченное время (с)**
Первая вставка данных	62.644
Вторая вставка данных	62.435
Третья вставка данных	61.983
Четвертая вставка данных	62.759
Среднее время выполнения	62.455

Таблица 2

Замер времени вставки РСУБД PostgreSQL

Операция	**Затраченное время (с)**
Первая вставка данных	325.354
Вторая вставка данных	326.753
Третья вставка данных	326.648
Четвертая вставка данных	326.045
Среднее время выполнения	326.2

При проведении данной операции очевидно преимущество СУБД MongoDB, так как времени понадобилось в 5 раз меньше, чем у РСУБД PostgreSQL.

2.2. Замер скорости выборки записей без использования индексов

Для сравнения быстродействия операций введем такой показатель как скорость (количество обработанных записей в секунду). Также одновременно с замером скорости, проводился замер процессорной нагрузки и объема потребляемой оперативной памяти.

Для замеров были взяты следующие запросы для СУБД MongoDB:

1. db.database.find({id:{}$gt:10000}) – сравнение целых чисел;
2. db.database.find({Level:/^E/}).explain() – поиск записей, в которых уровень логирования начинается с буквы «E»;
3. db.database.find({Text:/Error/}).explain() – поиск записей, в текст которых, входит слово «Error».
4. db.database.find({ConsumedMemory: {$lt::3000}}) – сравнение дробных чисел.

Для замеров в СУБД PostgreSQL были написаны аналоги запросов в синтаксисе SQL:

1. select * from database where id>10000;
2. select * from database where Level like ‘E%’;
3. select * from database where Text like ‘%Error%’;
4. select * from database where ConsumedMemory < 3000.

На рис. 1 представлен результат сравнения скорости операций в коллекции/таблице.

Высокая скорость операций в MongoDB обусловлено более активным использованием процессорного времени. Средние значения использования ресурсов компьютера представлены в табл. 3.

Таблица 3

Средние значения использования ресурсов компьютера

	MongoDB	PostgreSQL
CPU	30-40%	5-25%
RAM	700-800 мб.	500-1000 мб.

Рис. 1. Диаграмма скорости обработки данных без использования индексов

2.3. Замер времени вставки записей с использованием индексов

Для следующих замеров были созданы индексы на полях id и ConsumedMemory и сделана вставка 4 частей по 250 тысяч записей. Результаты замера приведены в табл. 4 и табл. 5.

Таблица 4

Замер времени вставки СУБД MongoDB с использованием индексов

Операция	**Затраченное время (с)**
Первая вставка данных	69.178
Вторая вставка данных	71.301
Третья вставка данных	69.421
Четвертая вставка данных	70.684
Среднее время выполнения	70.146

Таблица 5

Замер времени вставки РСУБД PostgreSQL с использованием индексов

Операция	**Затраченное время (с)**
Первая вставка данных	456.157
Вторая вставка данных	449.752
Третья вставка данных	451.602
Четвертая вставка данных	454.547
Среднее время выполнения	454.014

2.4. Замер скорости выборки записей с использованием индексов

Для замеров были взяты следующие запросы для СУБД MongoDB:

1. db.database.find({$and:[{id:{$gt:10000}},{id:{$lt:100000}}]})
2. db.database.find({$and:[{ConsumedMemory: {$lt:1000}},{ConsumedMemory: {$gt:5700}}]})

Для замеров в СУБД PostgreSQL были написаны аналоги запросов в синтаксисе SQL:

1. select * from database where id>10000 and id<100000;
2. select * from database where ConsumedMemory<1000 and ConsumedMemory >5700;

На рис. 2 представлен результат сравнения скорости операций в коллекции/таблице с использованием индексов.

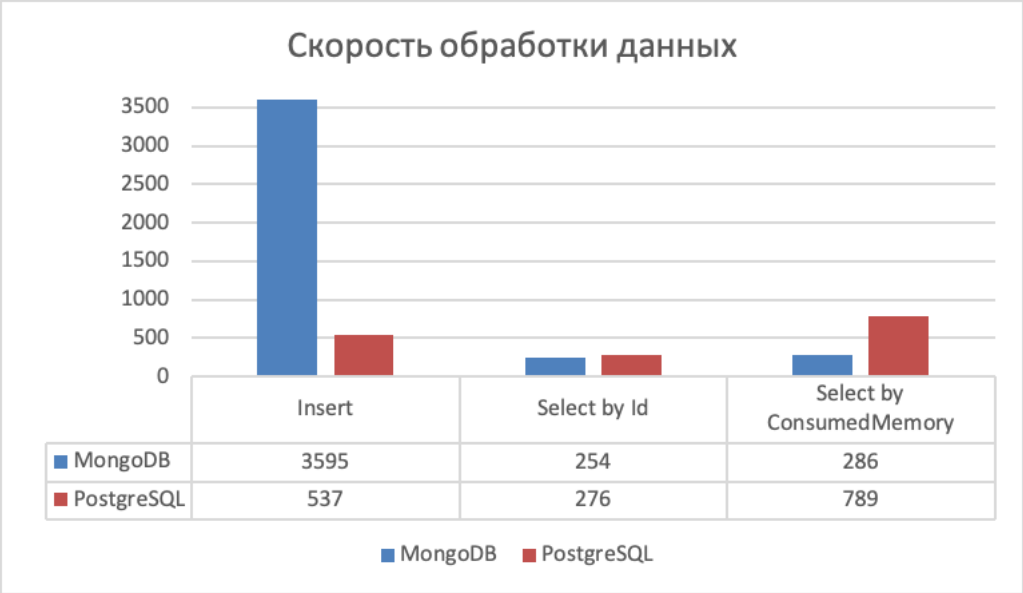


Рис. 2. Диаграмма скорости обработки данных с использованием индексов

Средние значения использования ресурсов компьютера представлены в табл. 6.

Таблица 6

Средние значения использования ресурсов компьютера

	MongoDB	PostgreSQL
CPU	1-2%	4-14%
RAM	650-1000 мб.	500-1000 мб.

Заключение

В данной работе было произведено тестирование реляционной системы управления данными PostgreSQL и NoSQL-системы MongoDB. Анализируя результат работы, можно видеть, что большое преимущество при вставке записей имеет СУБД MongoDB. При выборке из коллекции без индексов MongoDB также имеет преимущество. При тестировании с индексами скорость выборки в PostgreSQL стала больше, чем у MongoDB. Стоит заметить, что с использованием индексов у обоих систем уменьшилось потребление ресурсов компьютера. При выборе СУБД стоит исходить от конкретной задачи и расставлять приоритеты, что важнее - скорость вставки или скорость выборки, а также учитывать влияние использования индексов для ускорения запросов выборки.

Литература

1. *Kyle, B.* MongoDB in Action / B. Kyle. – 2-е изд., перераб. и доп. – New York: Manning Publications, 2016. – 480 с.

2. *Abramova Veronika, Bernardino Jorge, Furtado Pedro.* Which NoSQL Database? A Performance Overview. – 2014.

3. MongoDB. – Режим доступа свободный. – https://www.mongodb.org/. – (Дата обращения: 10.03.2021).

4. Замеры производительности на Java с JMH. – Режим доступа свободный. – https://nuancesprog.ru/p/8792/. – (Дата обращения: 12.04.2021).

5. PostgreSQL. – Режим доступа свободный. – https://postgresql.org. – (Дата обращения: 12.04.2021).

Строгонов Вячеслав Геннадьевич – магистрант 1-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: 432032@mail.ru

Борисенков Дмитрий Васильевич (научный руководитель) – канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: xuser@relex.ru

УДК 656.022

ОБ ОДНОМ АЛГОРИТМЕ РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ SDVRP ТИПА

Д. А. Татаренко

Воронежский государственный университет

Введение

В настоящее время рынок грузоперевозок состоит из большого числа транспортных компаний. Перевозка грузов представляет собой цепочку последовательных процессов, которые должны обеспечить максимально выгодные условия его перемещения. Главным фактором, определяющим величину транспортных расходов, является расстояние. От него зависит величина переменных издержек, таких как затраты на оплату труда водителю, затраты на топливо, затраты на техническое обслуживание автомобиля и других. Минимизации затрат и, как следствие, увеличению общей прибыли компании способствует качественное управление грузоперевозками. Ключевой задачей управления перевозками является составление наиболее оптимальных маршрутов следования транспортных средств. Задача маршрутизации транспортных средств является одним из основных классов задач транспортной логистики.

1. Классическая задача маршрутизации

Классическая задача маршрутизации (Vehicle Routing Problem, VRP) основывается на одной из основных задач комбинаторной оптимизации – задаче коммивояжера (Traveling Salesmen Problem, TSP), в которой требуется найти самый оптимальный путь, проходящий через все указанные города хотя бы по одному разу, с последующим возвратом в исходный город. VRP задача формулируется следующим образом: имеется автопарк однотипных транспортных средств, расположенный на складе (в депо). Есть потребители (клиенты) для каждого из которых известен спрос на доставку. Склад и потребители объединены в некоторую транспортную сеть. Расстояние между всеми пунктами сети считается известным. Необходимо найти множество маршрутов, начинающихся и заканчивающихся в депо, при условии, что все клиенты должны быть обслужены и каждый клиент посещается один раз. Однако на практике зачастую появляются дополнительные условия, накладывающие ограничения на классическую задачу маршрутизации. Выделяют следующие подклассы задачи маршрутизации [3, 4]:

– задача маршрутизации транспортных средств с учетом грузоподъемности (Capacitated VRP, CVRP) – предполагается, что каждое транспортное средство имеет ограниченную грузоподъемность;

– задача маршрутизации транспортных средств с временными окнами (VRP with Time Windows, VRPTW) – предполагается, что каждый клиент должен быть обслужен в определенный промежуток времени. Используется при ограниченном временном диапазоне приема или вывоза продукции потребителями;

– задача маршрутизации транспортных средств с множеством депо (Multiple Deport VRP, MDVRP) – для обслуживания клиентов используется несколько депо;

– периодическая задача маршрутизации транспортных средств (Periodic VRP, PVRP) – планирование работы транспорта ведется на период (несколько дней, неделя и т.д.). когда клиенты посещаются с разной частотой, заданной для каждого из них;

– задача маршрутизации транспортных средств с вывозом и доставкой (VRP with Pick-up and Delivery, VRPPD) – предполагается, что клиенты могут возвращать часть товаров обратно на склад;

– задача маршрутизации транспортных средств с возможностью дозагрузки (VRP with Satellite Facilities, VRPSF) – предполагается, что существует возможность дозагрузки транспортного средства на маршруте при помощи дополнительных складов;

– задача маршрутизации транспортных средств со случайными данными (Stochastic VRP) – предполагается, что некоторые компоненты задачи могут иметь случайное поведение. Используется в случае неизвестных или непостоянных значений спроса, количества состава парка, группы потребителей и т.д.;

– задача маршрутизации транспортных средств с раздельной доставкой (Split Delivery VRP, SDVRP) – в задаче снимается ограничение на посещение клиента только один раз, т.е. один клиент может быть обслужен одновременно несколькими транспортными средствами. Применяется для парка транспортных средств различной грузоподъемности.

2. Задача маршрутизации с раздельной доставкой (SDVRP)

2.1. Описание задачи маршрутизации SDVRP типа

Задача маршрутизации с раздельной доставкой, являясь расширением классической VRP задачи, позволяет обслуживать одного клиента несколькими транспортными средствами различной грузоподъемности. Данная задача возникает в случае, когда суммарный спрос всех клиентов превышает общую грузоподъемность всех транспортных средств автопарка и приходится заказ некоторых клиентов разбивать на несколько рейсов.

Преимущества задачи SDVRP по отношению к классической задаче VRP заключается в следующем:

– общий объем поставки разделяется на партии, каждая из которых отвозится потребителю соответствующим маршрутом. За счет разбиения могут сокращаться транспортные расходы;

– повышается степень загрузки подвижного состава;

– снижается уровень запасов у потребителей.

Недостатки задачи SDVRP заключаются в следующем:

– сложность планирования рейсов, особенно с дополнительными ограничениями по времени, т. к. весь подвижный состав не всегда находится в депо, часть транспортных средств может быть задействована на других маршрутах;

– неопределенность разбивки объема поставки на отдельные партии завоза каждому клиенту.

Постановка SDVRP задачи схожа с VRP задачей со следующими уточнениями: автопарк состоит из транспортных средств различной грузоподъемности и снимается ограничение на посещение потребителя только один раз.

2.2. Алгоритм решения SDVRP задачи

Задача составления оптимальных маршрутов доставки может быть решена в два этапа. На рис. 1 представлена блок-схема алгоритма решения SDVRP задачи.

В блок-схеме используются следующие обозначения:

n – число клиентов;

i – порядковый номер клиента;

Q – грузоподъемность автопарка;

b_i – спрос i-го клиента;

$inc(i)$ – оператор увеличения переменной i на 1.

На первом этапе производится поиск кратчайших расстояний между пунктами транспортной сети. Для этого предлагается использовать алгоритм Флойда [1], который основывается на

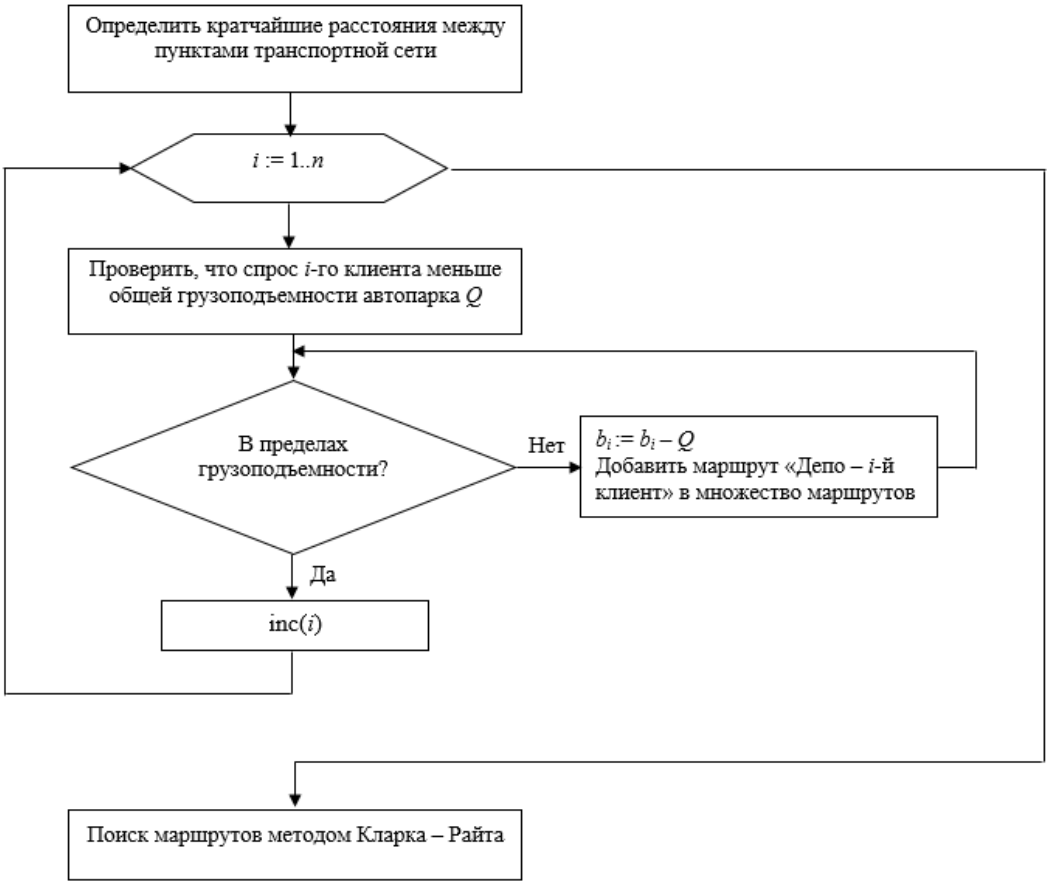


Рис. 1. Блок-схема решения SDVRP задачи

применении треугольного оператора, основная идея которого заключается в проверке того, не окажется ли путь из вершины i в вершину j короче, если будет проходить через некоторую промежуточную вершину k.

На втором этапе осуществляется поиск множества оптимальных маршрутов с использованием метода Кларка – Райта [2]. Однако, следует учесть, что данный алгоритм содержит в себе условие по ограничению суммарного объема поставок по маршруту: он не должен превышать общую грузоподъемность транспортного средства. Поэтому предварительно перед применением метода Кларка – Райта следует провести проверку, что спрос каждого клиента не превышает общую грузоподъемность всех транспортных средств автопарка. Если же для какого-нибудь клиента данное условие не выполняется, то следует разделить этот запрос. Тогда спрос клиента уменьшается на суммарную грузоподъемность автопарка и считается, что найден новый маршрут, состоящий из депо и указанного клиента. Необходимо выполнять данную проверку до тех пор, пока спрос каждого клиента не станет меньше суммарной грузоподъемности автопарка.

2.3. Пример решения SDVRP задачи

Рассмотри пример решения SDVRP задачи на примере транспортной сети (рис. 2).

Располагаемый парк транспортных средств имеет общую грузоподъемность 12 ед. Известен спрос каждого из четырех клиентов: спрос первого равен 5 ед., второго – 3 ед., третьего – 7 ед., четвертого – 14 ед.

Построим матрицу кратчайших расстояний между всеми узлами транспортной сети с использованием алгоритма Флойда. Полученный результат представлен в табл. 1.

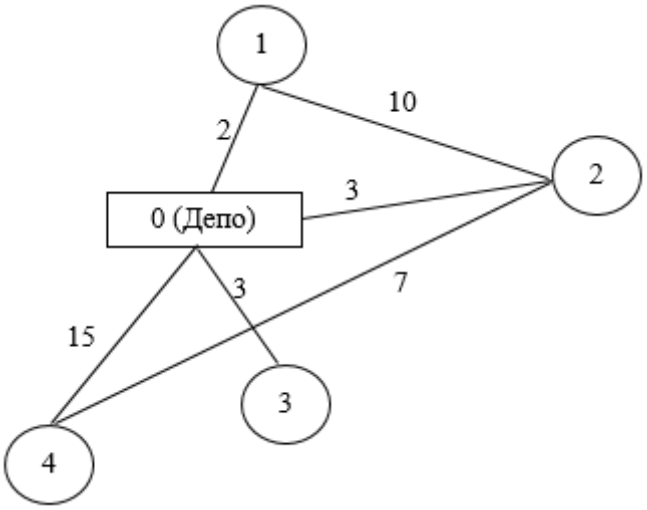


Рис. 2. Транспортная сеть

Таблица 1

Матрица кратчайших расстояний

	0	**1**	**2**	**3**	**4**
0	–	2	3	3	10
1	2	–	5	5	12
2	3	5	–	6	17
3	3	5	6	–	13
4	10	12	7	13	–

Проверим, что спрос каждого клиента не превышает грузоподъемность автопарка. Для 4-го клиента данное условие нарушено, следовательно, получен первый маршрут доставки 0 – 4 – 0 и спрос клиента становится равен 2.

Далее рассчитаем матрицу километровых выигрышей. Результат расчета представлен в табл. 2.

Таблица 2

Матрица километровых выигрышей

	0	**1**	**2**	**3**	**4**
0	–	2	3	3	10
1	0	–	5	5	12
2	0	0	–	6	17
3	0	0	0	–	13
4	0	0	6	0	–

Определим маршруты с использованием метода Кларка – Райта. В результате получены следующие маршруты: 0 – 4 – 2 – 1 – 0 и 0 – 3 – 0. Таким образом, в общем итоге получено три маршрута. При определении конечных маршрутов необходимо использовать кратчайший путь, т.е. с учетом матрицы кратчайших расстояний, например, для маршрута 0 – 4 – 0 получается итоговый маршрут 0 – 2 – 4 – 2 – 0.

Заключение

Эффективное управление грузоперевозками предполагает поиск оптимальных решений еще на этапе планирования перевозок для сокращения пробега транспорта и снижения затрат. В данной работе была рассмотрена задача маршрутизации транспортных средств с раздельной доставкой и предложен алгоритм ее решения. Указанный алгоритм позволяет отыскать оптимальные маршруты при минимизации общего пройденного расстояния.

Дальнейшим развитием полученного решения SDVRP задачи является распределение транспортных средств по маршрутам следования по критерию минимизации себестоимости рейса.

Введем следующие обозначения:

S – число типов транспортных средств, имеющихся в автопарке;

L – количество полученных ранее маршрутов;

c_{sl} – себестоимость одного рейса, выполняемого транспортным средством s-го типа на l-м маршруте;

N_s – количество транспортных средств s-го типа в автопарке фирмы;

q_s – грузоподъемность транспортного средства s-го типа;

O_l – объем перевозок груза по l-му маршруту, определенный при поиске маршрутов;

y_{sl} – искомое количество транспортных средств, назначаемых на l-й маршрут.

Тогда задача распределения транспортных средств для каждого отдельно взятого маршрута l имеет следующий вид:

$$\sum_{s=1}^{S} c_{sl} y_{sl} \to \min, l = 1..L$$

$$\sum_{s=1}^{S} q_s y_{sl} \geq O_l$$

$$\sum_{l=1}^{L} y_{sl} \leq N_s, s = 1...S$$

$$y_{sl} = [y_{sl}] \geq 0$$

Данная задача относится к задачам целочисленного линейного программирования и может быть решена методом ветвей и границ.

Литература

1. *Асанов, М. О.* Дискретная математика: графы, матроиды, алгоритмы: учеб. пособие / М. О. Асанов, В. А. Баранский, В. В. Расин. – 2-е изд., испр. и доп. – Санкт-Петербург : Лань, 2010. – 368 с.

2. *Черкесов, А. Г.* Экономика: практические задачи и решения: учеб. пособие / А. Г. Черкесов. – Санкт-Петербург : Изд-во СПбГТУ, 2002. – 50 с.

3. *Тюрин, А. Ю.* Модели транспортного обслуживания в цепях поставок пищевой промышленности / А. Ю. Тюрин // Вестник Кузбасского гос. технич. ун-та. – 2011. – № 4(86). – С. 89–92.

4. *Маслеев, А. И.* Тестирование эвристики для решения задачи маршрутизации транспорта с раздельной доставкой / А. И. Маслеев, А. Д. Кулязин, А. В. Липенков // Транспортный системы. – 2019. – №4(14). – С. 26–33.

Татаренко Дарья Алексеевна – магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: dtatarienko@bk.ru

Булгакова Ирина Николаевна (научный руководитель) – д-р экон. наук, доц., доцент кафедры системного анализа и управления Воронежского государственного университета.
E-mail: bulgakova-i-n@yandex.ru

УДК 004.932

ПРИМЕНЕНИЕ МЕТОДА ДЕКОМПОЗИЦИОННОГО ДЕРЕВА ДЛЯ ВЫДЕЛЕНИЯ ОЪЕКТОВ НА ИЗОБРАЖЕНИИ

Е. А. Тихомирова

Воронежский государственный университет

Введение

Одной из основных задач, решаемых при обработке изображения, является задача сегментации изображения. На сегодняшний день существует множество различных методов сегментации [1–3]. Цель сегментации заключается в упрощении или изменении представления изображения, чтобы его было проще и легче анализировать, а также значительно снизить время обработки изображения, так как полученное представление значительно уменьшает количество примитивов по сравнению с пиксельным.

Задача сегментации очень схожа по постановке с другой часто решаемой задачей анализа данных – задачей кластеризации. Тогда в качестве признаков точки изображения можно использовать представление ее цвета в некотором цветовом пространстве, примером меры близости может быть евклидово расстояние между данными векторами.

Определенный интерес среди методов решения задачи кластеризации представляют методы, для которых исходная информация формируется в виде матрицы нечеткого отношения сходства. При обработке такой матрицы удается не только получить некоторое разбиение объектов на кластеры, но и получить альтернативные варианты разбиения на том же множестве. Одним из таких метод является метод *декомпозиционного дерева* [4–6], идея которого первоначально была изложена в [7]. Одним из основных преимуществ этого метода является возможность получения всех потенциальных разбиений исходного множества. Поэтому определенный интерес представляет возможность использования данного метода для кластеризации изображения.

Цель статьи заключается в рассмотрении результата применения метода декомпозиционного дерева для выделения объектов на изображении.

1. Суперпиксельная сегментация

На этапе предобработки изображение обычно преобразуют в набор суперпикселей. Данная процедура называется *суперпиксельной сегментацией*. *Суперпиксельная сегментация* изображения – разбиение изображения на непересекающиеся области или сегменты, покрывающие все изображение. Такие сегменты так же называются *суперпикселями*. Результатом такого разбиения является *суперпиксельная карта*. Сегментация позволяет упростить изображение для большего удобства при дальнейшем анализе.

Каждый суперпиксель представляет собой некоторый кластер, то есть объединение нескольких однородных в некотором смысле пикселей, например, по цвету, текстуре или яркости, при этом пиксели из соседних областей существенно отличаются. Таким образом в дальнейшей обработке каждый суперпиксель может быть рассмотрен как единое целое.

Полученные сегменты как правило имеют неправильную форму. Тем самым полученное разбиение позволяет лучше выделить границы отдельных объектов на изображении.

При уменьшении числа сегментов в суперпиксельной карте, увеличивается шанс, что сегмент будет покрывать разные объекты изображения (рис. 1). Так, например, алгоритмом SLIC достаточно сложно выделить целый объект и только его с помощью одного суперпикселя.

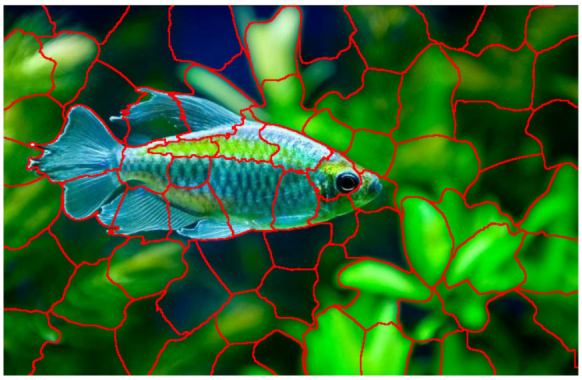


Рис. 1. Пример разбиения на 50 суперпикселей

2. Метод декомпозиционного дерева

Рассмотрим задачу нечеткой кластеризации. Пусть задано множество объектов X. Причем каждый объект $x \in X$ характеризуется набором признаков $V_i = \{v_{i1}, \ldots, v_{im}\}$, $i = \overline{1,n}$. Требуется определить классы ближайших объектов, то есть такие, которые попадут в один класс эквивалентности. Как правило, «близость» оценивается с помощью подходящей функции расстояния. В методе декомпозиционного дерева используются так называемые *транзитивные расстояния*. На рис. 2 представлена схема данного метода.

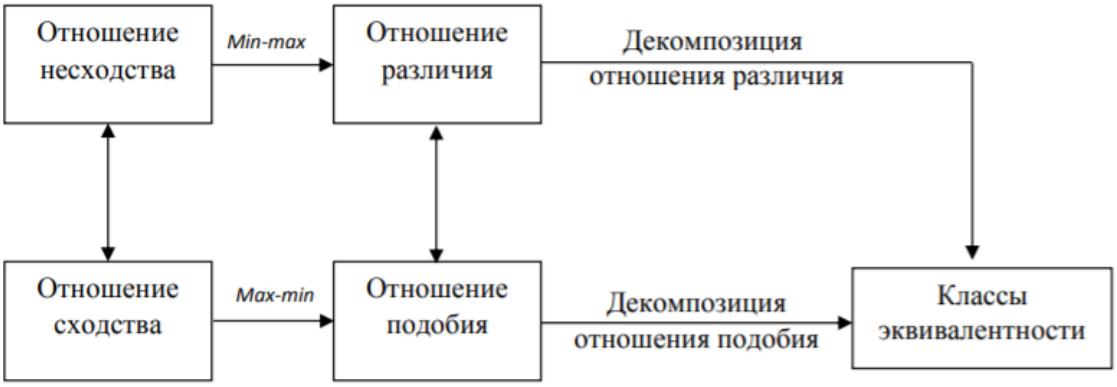


Рис. 2. Схема метода декопозиционного дерева

На основе исходного множества строится отношение сходства или несходства с помощью функции расстояния. Используя операцию дополнения несложно перейти от одного отношения к другому и наоборот. Так как отношения сходства и несходства не обладают свойством транзитивности, необходимо обеспечить это свойство. Для этого используется операция транзитивного замыкания. Полученное отношение называется отношением подобия (различия), оно обладает свойствами: рефлексивности (антирефлексивности), симметричности и транзитивности. Согласно *теореме декомпозиции* любой α-срез подобия является обычным отношением эквивалентности и, следовательно, определяет разбиение множества объектов на непересекающиеся классы эквивалентности. Наглядно такое отношение можно представить в виде схемы, которая называется *декомпозиционным деревом*.

Метод декомпозиционного дерева обладает рядом преимуществ:

– возможность работать с разнородной информацией за счет того, что формируемое на начальном шаге отношение несходства базируется на специальных функциях (индексы несходства, функция подобия и др.), область определения которых может быть произвольной, связанной со шкалой измерений, а область значений всегда можно представить в виде [0,1];

– декомпозиционное дерево позволяет проследить последовательный процесс разбиения исходного множества от кластера, совпадающего с исходным множеством объектов, до разбиения на кластеры, состоящие из одного элемента (тривиальное разбиение);

– использование параметрических нечетких операций (в частности, треугольных норм и конорм) позволяет осуществить настройку метода на конкретного пользователя или конкретную проблему.

Одним из основных преимуществ данного метода кластеризации является возможность получения всевозможных разбиений исходного множества.

3. Иллюстративный пример

Для сегментации изображения воспользуемся алгоритмом SLIC (Simple Linear Iterative Clustering) [1], который является одним из наиболее простых и распространённых алгоритмов суперпиксельной сегментации. Данный алгоритм является модификацией классического алгоритма кластеризации k-средних. Главное отличие от k-средних заключается в том, что данный метод ограничивает область поиска однородных пикселей для каждого сегмента. Каждая точка в таких кластерах характеризуется пятимерным вектором (l, a, b, x, y), где l, a, b – координаты цвета, а x, y – координаты пространства. А в качестве меры близости используется взвешенная сумма евклидовых расстояний.

В результате применения данного метода была получена суперпиксельная карта, состоящая из 97 сегментов (рис. 3).

Рис. 3. Сегментация изображения алгоритмом SLIC

Каждому суперпикселю S_i, $i = \overline{1,97}$ поставим в соответствие некоторый вектор признаков $C_i = \{r_i, g_i, b_i\}$, где r_i, g_i, b_i получены путем усреднения значений RGB по каждому пикселю, входящему в сегмент S_i.

Для классификации полученных сегментов воспользуемся методом декомпозиционного дерева. Для этого построим матрицу относительных евклидовых расстояний, которую будем рассматривать как матрицу нечеткого отношения несходства R. В соответствии с алгоритмом метода перейдем к матрице отношения сходства $\overline{R}$. Определив максиминное транзитивное замыкание, тем самым получив отношение подобия $\widehat{\overline{R}}$.

Каждое значение $\alpha \in \{0,85; 0,91; 0,92; 0,93; 0,94; 0,95; 0,96; 0,98; 1\}$ из полученной матрицы определяет отношение эквивалентности R_α, которое является α-срезом нечеткого отношения R. Полученные матрицы R_α приводят к блочно-диагональному виду, где каждый блок определяет класс эквивалентности. На рис. 4, 5 представлены результаты разбиения при раз-

личных $\alpha = 0,91; 0,92; 0,93; 0,94; 0,95$. Легко заметить, что при увеличении значений α классы эквивалентности могут только разбиваться на более мелкие.

Данных результатов вполне достаточно, чтобы понять, как примерно будут выглядеть разбиения для оставшихся α. Наилучшим будем считать разбиение при котором наиболее явно выделяются объекты на фоне окружающей их среды. Таким разбиениям соответствуют разбиения при $\alpha = 0,91; 0,94$.

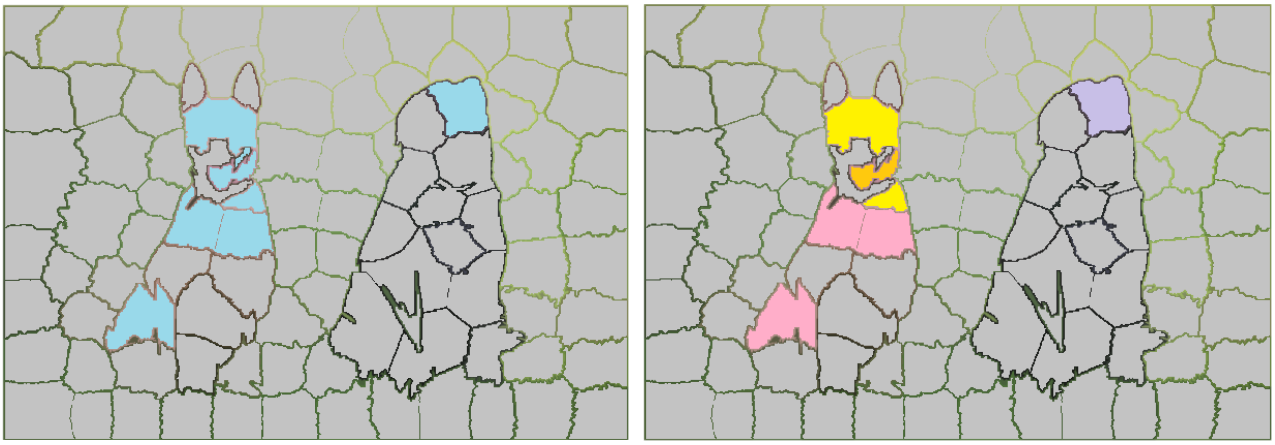


Рис. 4. Результат разбиения при $\alpha = 0,91; 0,92$

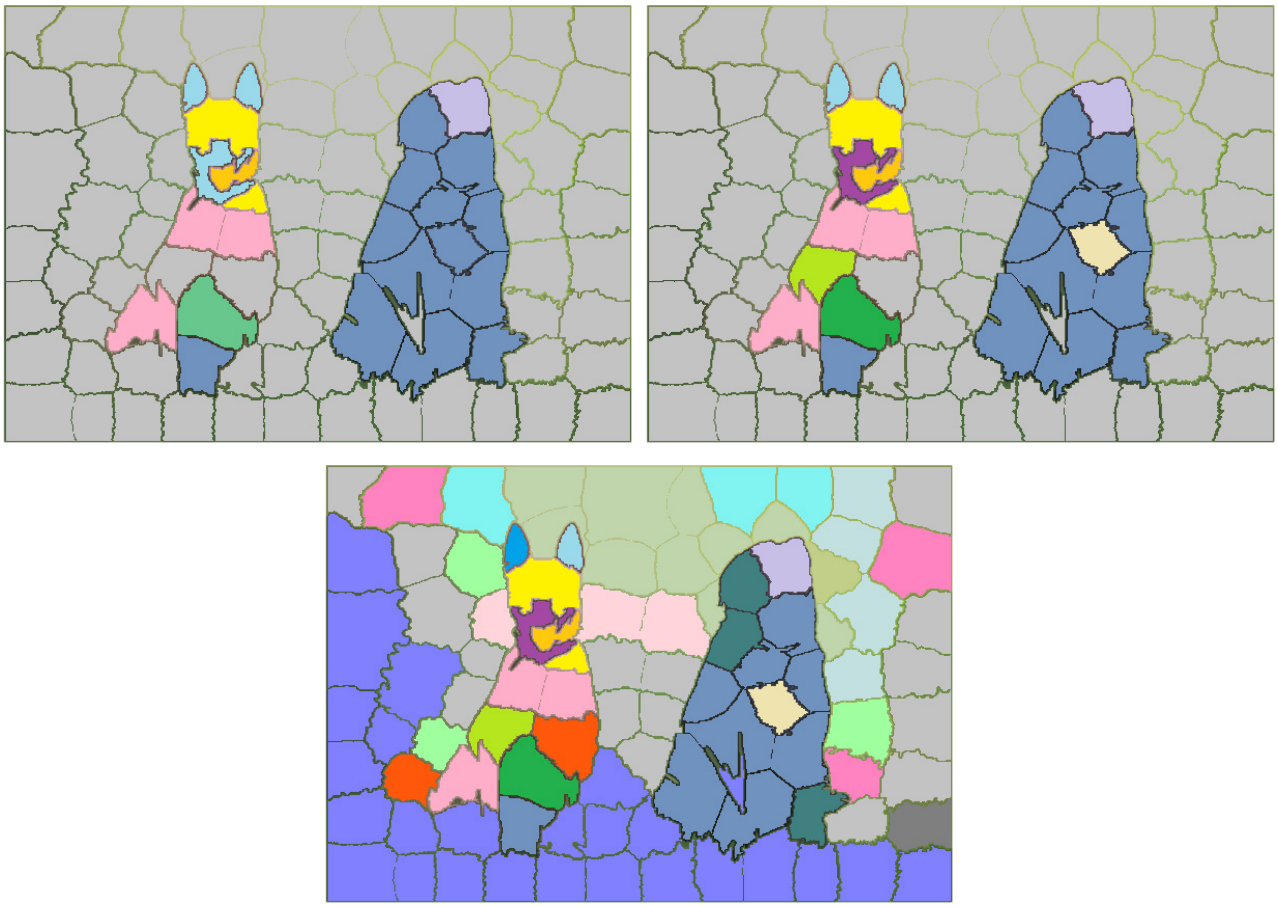


Рис. 5. Результат разбиения при $\alpha = 0,91; 0,94; 0,95$

Результат можно попробовать улучшить, например добавив в вектор признаков C_i каждого суперпикселя координаты его геометрического центра. Или же использовать процедуру выделения связных компонент на полученном разбиении.

Литература

1. *Achanta, R.* SLIC superpixels compared to state-of-the-art superpixel methods / R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk // IEEE Trans. Pattern Anal. – 2012. – Vol. 34, No 11. – P. 2274–2282.

2. *Wu, Z.* An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation / Z. Wu, R. Leahy// IEEE transactions on pattern analysis and machine intelligence. – 1993. – No 11.

3. *Felzenszwalb, P. F.* Efficient graph-based image segmentation / P. F. Felzenszwalb, D. P. Huttenlocher // International Journal of Computer Vision. – 2004. – P. 167–181.

4. *Леденева, Т. М.* Обработка нечеткой информации / Т. М. Леденева. Воронеж : Изд-во ВГУ, 2006. – 233 с.

5. *Палагина А. М.* Оценка качества разбиений в методе декомпозиционного дерева / А. М. Палагина, Н. А. Каплиева // Актуальные проблемы прикладной математики, информатики и механики: Сб. труд. Междунар. науч.-техн. конф. – 2018. – С. 252–256.

6. *Леденева, Т. М.* О влиянии функции подобия на результаты нечеткой классификации / Т. М. Леденева, Нгуен Нгок Хуи // Информационные технологии. – М. : Новые технологии, 2011. – № 11. – С. 15–23.

7. *Felzenszwalb, P. F.* Efficient graph-based image segmentation / P. F. Felzenszwalb, D. P. Huttenlocher // International Journal of Computer Vision. – 2004. – P. 167–181.

Тихомирова Екатерина Александровна – магистрант 1-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: katya246893@gmail.com

Леденёва Татьяна Михайловна (научный руководитель) – д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: ledeneva-tm@yandex.ru

УДК 539.3

НАПРЯЖЕННО-ДЕФОРМИРОВАННОЕ СОСТОЯНИЕ УПРОЧНЯЮЩЕГОСЯ УПРУГОВЯЗКОПЛАСТИЧЕСКОГО ТЕЛА ПОД ДЕЙСТВИЕМ ТЕМПЕРАТУРЫ В УСЛОВИЯХ СФЕРИЧЕСКОЙ СИММЕТРИИ

Д. А. Чернышов

Воронежский государственный университет

Введение

В данной работе рассмотрено решение задачи о напряженно-деформированном состоянии сплошного упрочняющегося упруговязкопластического шара при нагревании. Модель материала позволяет учитывать особенности деформирования материалов со сложной реологией. Целью работы является получение выражений для напряжений перемещений и границ раздела зон упругого, пластического деформирования и зоны разгрузки.

В ходе работы были поставлены и решены следующие задачи: получить аналитическое и графическое решение задачи; исследовать полученные зависимости.

Решению задач механики для материалов с реологическими свойствами посвящено большое количество книг и статей, например, [1, 4, 6, 7]. В работе [3] получено решение данной задачи в рамках предположения об упругопластичности материала, а в [2] приводится ее усложнение с учетом упрочнения.

1. Постановка задачи

Пусть сплошной шар радиуса R, имеет начальную температуру T_0. Он нагревается таким образом, что в каждый момент времени на поверхности выполнено условие $\frac{T(R,t)-T_0}{T_m-T_0}=1-\mathrm{e}^{-xt}$, где x – скорость нагрева, T_m – максимальная температура нагрева. Примем, что внешняя поверхность шара свободна от усилий, массовые силы отсутствуют. Будем считать, что шар сделан из упрочняющегося упруговязкопастического материала с изменяющимся пределом текучести. Определим напряженно-деформированное состояние тела в поставленной задаче.

2. Решение задачи в условиях сферической симметрии

2.1. Задача теплопроводности

В условиях сферической симметрии уравнение теплопроводности примет вид:

$$\frac{\partial T}{\partial t}=\vartheta\left(\frac{2}{r}\frac{\partial T}{\partial r}+\frac{\partial^2 T}{\partial r^2}\right).$$

Здесь ϑ – коэффициент температуропроводности.

Запишем начальное и граничные условия, а также условие симметрии:

$$T(r,0)=T_0$$

$$\frac{T(R,t)-T_0}{T_m-T_0}=1-\mathrm{e}^{-xt}$$

$$T(0,t)\neq\infty$$

$$\left.\frac{\partial T}{\partial r}\right|_{r=0} = 0.$$

Таким образом, получена замкнутая система уравнений. Решение аналогичной задачи, отличающейся видом граничного условия, приведено в [5, с. 105]. Воспользовавшись данным результатом, запишем решение в безразмерном виде для нашего случая:

$$\theta(\xi,t) = \frac{T(\xi,t)-T_0}{T_m-T_0} = \left(1-\mathrm{e}^{-xt}\right)\left(1+2\sum_{n=1}^{\infty}(-1)^n \frac{\sin(\pi n\xi)}{\pi n\xi} e^{-\frac{\pi^2 n^2}{R^2}\vartheta t}\right),$$

где ξ – безразмерная координата.

2.2. Напряженно-деформированное состояние

Основные соотношения

Приведем основные соотношения, используемые при решении задач механики сплошной среды, с учетом реологических свойств материала и температуры.

Уравнения равновесия:

$$\frac{\partial \sigma_r}{\partial r} + \frac{2}{r}\left(\sigma_r - \sigma_\varphi\right) = 0.$$

Будем рассматривать модель сложной среды, предложенной А. Н. Спорыхиным и учитывающую упругий, вязкий и пластический механизмы деформирования:

$$f\left(\sigma_r^a, \sigma_\varphi^a\right) = 0,$$

где

$$\sigma_r^a = \sigma_r - ce_r^p - \eta\dot{e}_r^p,$$

$$\sigma_\varphi^a = \sigma_\varphi - ce_\varphi^p - \eta\dot{e}_\varphi^p.$$

Здесь c – коэффициент упрочнения, η – коэффициент вязкости.

В качестве условия пластичности выберем условие Треска:

$$\left|\left(\sigma_r - ce_r^p - \eta\dot{e}_r^p\right) - \left(\sigma_\varphi - ce_\varphi^p - \eta\dot{e}_\varphi^p\right)\right| = 2k(r,t).$$

Будем рассматривать материал, для которого предел текучести линейно зависит от температуры:

$$k(r,t) = k_0\left(1-\gamma\Delta(r,t)\right).$$

где k_0 – предел текучести при комнатной температуре, γ – некоторая константа, определяемая из экспериментов, $\Delta(r,t) = \alpha\left(T(r,t)-T_0\right)$, α – коэффициент температурного расширения.

Условие пластической несжимаемости материала:

$$e_r^p + 2e_\varphi^p = 0.$$

Закон Дюамеля – Неймана:

$$e_r^e = \Delta(r,t) + \frac{(\lambda+\mu)\sigma_r - \lambda\sigma_\varphi}{\mu(3\lambda+2\mu)},$$

$$e_\varphi^e = e_\theta^e = \Delta(r,t) + \frac{(\lambda+2\mu)\sigma_\varphi - \lambda\sigma_r}{2\mu(3\lambda+2\mu)}.$$

Уравнения для полных деформаций и соотношения Коши:

$$e_r^e + e_r^p = e_r = u_{r,r},$$

$$e_\varphi^e + e_\varphi^p = e_\theta^e + e_\theta^p = e_\varphi = e_\theta = \frac{u_r}{r}.$$

Упругая зона

В начальный момент времени тело находится в состоянии упругого равновесия. Поэтому для упругого деформирования во всем шаре решение записывается в виде

$$\sigma_r = -\frac{4\omega}{r^3}\int_0^r \rho^2\Delta(\rho,t)\,\mathrm{d}\rho + A_1(t) + \frac{B_1(t)}{r^3},$$

$$\sigma_\varphi = \frac{2\omega}{r^3}\int_0^r \rho^2\Delta(\rho,t)\,\mathrm{d}\rho - 2\omega\Delta(r,t) + A_1(t) - \frac{B_1(t)}{2r^3},$$

$$u = \frac{\omega}{\mu r^2}\int_0^r \rho^2\Delta(\rho,t)\,\mathrm{d}\rho + \frac{rA_1(t)}{3\lambda+2\mu} - \frac{B_1(t)}{4\mu r^2},$$

где $\omega = \dfrac{\mu(3\lambda+2\mu)}{\lambda+2\mu}$.

Функции интегрирования определяются из условий свободной от усилий поверхности и отсутствия перемещений в центре.

Зона пластичности

В некоторый момент $t = t_p$ на поверхности шара $r = R$ начинает выполняться условие пластичности

$$\left(\sigma_r - ce_r^p - \eta\dot{e}_r^p\right) - \left(\sigma_\varphi - ce_\varphi^p - \eta\dot{e}_\varphi^p\right) = 2k(r,t).$$

В связи с этим при $t > t_p$ в шаре одновременно существуют область обратимого ($0 \le r < a(t)$) и необратимого ($a(t) \le r \le R$) деформирования. $a(t)$ – упругопластическая граница.

В пластической зоне решение описывается соотношениями

$$\sigma_r = -\frac{8\omega}{3\eta}\mathrm{e}^{-\varkappa t}\int_{t_p}^t\int_a^r \frac{\mathrm{e}^{\varkappa\tau}k(\rho,\tau)}{\rho}\,\mathrm{d}\rho\,\mathrm{d}\tau + \frac{8\omega^2}{3\eta}\frac{\mathrm{e}^{-\varkappa t}}{r^3}\int_{t_p}^t\int_a^r \mathrm{e}^{\varkappa\tau}\rho^2\Delta(\rho,\tau)\,\mathrm{d}\rho\,\mathrm{d}\tau - \frac{4\omega}{r^3}\int_a^r \rho^2\Delta(\rho,t)\,\mathrm{d}\rho +$$

$$+ A_2(t) + \frac{B_2(t)}{r^3} - \frac{2\omega}{3\eta}\frac{\mathrm{e}^{-\varkappa t}}{r^3}\int_{t_p}^t \mathrm{e}^{\varkappa\tau}B_2(\tau)\,\mathrm{d}\tau + 2\omega\mathrm{e}^{-\varkappa t}\int_a^r \frac{C_2(\rho)}{\rho}\,\mathrm{d}\rho,$$

$$\sigma_\varphi = -\frac{8\omega}{3\eta}\mathrm{e}^{-\varkappa t}\int_{t_p}^t\int_a^r \frac{\mathrm{e}^{\varkappa\tau}k(\rho,\tau)}{\rho}\,\mathrm{d}\rho\,\mathrm{d}\tau - \frac{4\omega^2}{3\eta}\frac{\mathrm{e}^{-\varkappa t}}{r^3}\int_{t_p}^t\int_a^r \mathrm{e}^{\varkappa\tau}\rho^2\Delta(\rho,\tau)\,\mathrm{d}\rho\,\mathrm{d}\tau + \frac{2\omega}{r^3}\int_a^r \rho^2\Delta(\rho,t)\,\mathrm{d}\rho -$$

$$- \frac{4\omega}{3\eta}\mathrm{e}^{-\varkappa t}\int_{t_p}^t \mathrm{e}^{\varkappa\tau}k(r,\tau)\,\mathrm{d}\tau + \frac{4\omega^2}{3\eta}\mathrm{e}^{-\varkappa t}\int_{t_p}^t \mathrm{e}^{\varkappa\tau}\Delta(r,\tau)\,\mathrm{d}\tau - 2\omega\Delta(r,t) + A_2(t) - \frac{B_2(t)}{2r^3} +$$

$$+ \frac{\omega}{3\eta}\frac{\mathrm{e}^{-\varkappa t}}{r^3}\int_{t_p}^t \mathrm{e}^{\varkappa\tau}B_2(\tau)\,\mathrm{d}\tau + 2\omega\mathrm{e}^{-\varkappa t}\int_a^r \frac{C_2(\rho)}{\rho}\,\mathrm{d}\rho + \omega\mathrm{e}^{-\varkappa t}C_2(r),$$

$$u = -\frac{8\omega}{3\eta(3\lambda+2\mu)}\mathrm{e}^{-\varkappa t}r\int_{t_p}^t\int_a^r \frac{\mathrm{e}^{\varkappa\tau}k(\rho,\tau)}{\rho}\,\mathrm{d}\rho\,\mathrm{d}\tau + \frac{8\omega^2}{3\eta(3\lambda+2\mu)}\frac{\mathrm{e}^{-\varkappa t}}{r^2}\int_{t_p}^t\int_a^r \mathrm{e}^{\varkappa\tau}\rho^2\Delta(\rho,\tau)\,\mathrm{d}\rho\,\mathrm{d}\tau +$$

$$+ \frac{\omega}{\mu r^2}\int_a^r \rho^2\Delta(\rho,t)\,\mathrm{d}\rho + \frac{rA_2(t)}{3\lambda+2\mu} - \frac{B_2(t)}{4\mu r^2} - \frac{2\omega}{3\eta(3\lambda+2\mu)}\frac{\mathrm{e}^{-\varkappa t}}{r^2}\int_{t_p}^t \mathrm{e}^{\varkappa\tau}B_2(\tau)\,\mathrm{d}\tau +$$

$$+ \frac{2\omega}{3\lambda+2\mu}\mathrm{e}^{-\varkappa t}r\int_a^r \frac{C_2(\rho)}{\rho}\,\mathrm{d}\rho,$$

где $\varkappa = \dfrac{2\omega+3c}{3\eta}$.

Функции интегрирования для обоих зон находятся из граничных и начальных (отсутствие на поверхности в момент $t = t_p$ пластических деформаций) условий, а также условий сопряжения на границе $a(t)$.

Зона разгрузки

По мере нагревания температурные напряжения в теле уменьшаются. Это приводит к снижению уровня пластической деформации в зоне пластического течения. При $t > t_u$ на поверхности шара возникает зона разгрузки с границей $b(t)$. В каждой точке $a(t) \le r \le R$ сохраняется уровень накопленной деформации $\hat{e}_{ij}^p(r)$. Он определяется пластической деформации в момент прохождения через точку границы $b(t)$.

Выражения в зоне разгрузки имеют вид

$$\sigma_r = -\frac{4\omega}{r^3}\int_b^r \rho^2 \Delta(\rho,t)\,\mathrm{d}\rho + 2\omega\int_b^r \frac{\hat{e}_r^p(\rho)}{\rho}\,\mathrm{d}\rho + A_3(t) + \frac{B_3(t)}{r^3},$$

$$\sigma_\varphi = \frac{2\omega}{r^3}\int_b^r \rho^2 \Delta(\rho,t)\,\mathrm{d}\rho + 2\omega\int_b^r \frac{\hat{e}_r^p(\rho)}{\rho}\,\mathrm{d}\rho + 2\omega\hat{e}_r^p(r) - 2\omega\Delta(r,t) + A_3(t) - \frac{B_3(t)}{2r^3},$$

$$u = \frac{\omega}{\mu r^2}\int_b^r \rho^2 \Delta(\rho,t)\,\mathrm{d}\rho + \frac{2\omega}{3\lambda+2\mu} r\int_b^r \frac{\hat{e}_r^p(\rho)}{\rho}\,\mathrm{d}\rho + \frac{rA_3(t)}{3\lambda+2\mu} - \frac{B_3(t)}{4\mu r^2}.$$

Теперь вновь необходимо воспользоваться граничными условиями и условием непрерывности радиальных напряжений и перемещений на упругопластических границах и определить из них функций интегрирования.

Зона повторной пластичности

При продолжении нагревания в зависимости от уровня накопленной деформации в момент $t > t_r$ возможен выход на условие повторной пластичности с противоположным знаком. Решение в новой области $c(t) \le r \le R$ совпадает с решением в $a(t) \le r < b(t)$ с точностью до знака перед пределом текучести, момента возникновения зоны и новых функций интегрирования.

Для определения функций интегрирования во всех зонах, как и прежде, будем использовать граничные условия, условия сопряжения на упругопластических границах, отсутствия пластических деформаций в момент начала первого пластического течения, а также равенство накопленных и новых пластических деформаций на границе $c(t)$.

Заключение

В работе получено аналитическое решение задачи о нагреве сплошного шара для зон упругости, пластичности, разгрузки и повторной пластичности; построены графические зависимости температуры и напряжений для материалов с различными свойствами.

При совершении предельного перехода $c = \eta = 0$ выражения для напряжений и перемещений, а также графические зависимости принимают вид, приведенный в [3].

Литература

1. *Артемов, М. А.* Об алгоритмах расчёта термопластического состояния диска / М. А. Артемов, Е. С. Барановский, Г. Г. Бердзенишвили // Ученые записки Комсомольского-на-Амуре государственного университета. – 2018. – Т. 1, № 3. – С. 25–30.

2. *Горностаев, К. К.* Задачи определения температурных напряжений и перемещений в сферических и цилиндрических телах со сложной реологией: дис… канд. физ.-мат. наук. – Воронеж, 2020. – с. 15–50.

3. *Дац, Е. П.* Неустановившиеся температурные напряжения в условиях зависимости предела текучести от температуры : дис… канд. физ.-мат. наук. – Владивосток, 2017. – с. 23–48.

4. *Качанов, Л. М.* Основы теории пластичности / Л. М. Качанов – Москва : Наука, 1969. – 420 с.

5. *Лыков, А. В.* Теория теплопроводности / А. В. Лыков. – Москва : Высшая школа, 1967. – 600 с.

6. *Паркус, Г.* Неустановившиеся температурные напряжения : пер. с нем. – Москва : Физматгих, 1963. – 252 с.

7. *Спорыхин, А. Н.* Неодномерные задачи упруговязкопластичности с неизвестной границей / А. Н. Спорыхин, А. В. Ковалев, Ю. Д. Щеглова : Воронеж. гос. ун-т .– Воронеж : Воронеж. гос. ун-т, 2004 .– 218 с.

Чернышов Данил Алексеевич – студент 4-го курса кафедры механики и компьютерного моделирования Воронежского государственного университета.
E-mail: chernyshov.danil@gmail.com

Ковалев Алексей Викторович (научный руководитель) – д-р физ.-мат. наук, проф., заведующий кафедрой механики и компьютерного моделирования Воронежского государственного университета. E-mail: kav-mail@mail.ru

УДК 004.932.72.1

СЕГМЕНТАЦИЯ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ МЕТОДА К-СРЕДНИХ И МОДИФИЦИРОВАННОГО МЕТОДА ВОДОРАЗДЕЛА

Д. Е. Черняев

Воронежский государственный университет

Введение

Сегментация изображений является неотъемлемой частью множества задач, связанных с медициной, в частности, с анализом медицинских изображений. Качество изображения влияет на визуализацию той области на изображении, которая является основой для постановки медицинского диагноза или планирования для предстоящей операции [2]. Метод водораздела является одним из часто используемых методов для сегментации медицинских изображений. Название алгоритма произошло от математической морфологии, которая имеет дело с топографическим представлением изображения [3, 4]. Набор пикселей с наименьшей региональной отметкой соответствует региональному минимуму. Региональный минимум – это множество пикселей со строго более низким уровнем интенсивности серого цвета, чем у соседних по отношению к ним пикселей. Метод водораздела базируется на принципе дождевого моделирования [3]. Как только начинает идти дождь, то любая капля дождя, достигая поверхности, будет течь по самому крутому спуску, пока не достигнет минимума. Пути пикселей, которые сходятся к общему минимуму, составляют водосборную полость (бассейн). Водоразделы – это возвышенности, разделяющие разные бассейны. Области, которые мы намереваемся получить путём кластеризации являются этими бассейнами, а их границы – водоразделы. Преимуществом метода водораздела является то, что это быстрый, простой и интуитивно понятный метод. Более того, что немаловажно, он может производить полное разбиение изображения на области, поэтому нет необходимости выполнять какую-либо постобработку, например, стыковку контуров. Однако, его недостатками являются чрезмерная сегментация и чувствительность к световому шуму [5].

Нельзя не упомянуть алгоритмы «мягкой» сегментации, активно использующиеся в сегментации магнитно-резонансных (далее МР) изображений [3, 4], где пиксели могут быть частично разделены на несколько однородных групп. Например, широко используется метод нечеткой кластеризации c-средних при сегментации МРТ изображений [6]. Однако, к его основным недостаткам можно отнести вычислительную сложность и тот факт, что производительность значительно ухудшается с увеличением светового шума. В тоже время, метод k-средних, является достаточно простым методом кластеризации с низкой вычислительной сложностью по сравнению с нечётким *c*-средних. Области, полученные данным методом, не пересекаются друг с другом. [7]

В данной статье рассмотрены примеры совместного использования метода k-средних при первичной сегментации изображения с последующим применением к полученным результатам метода водораздела.

1. Кластеризация методом k-средних

Мы используем метод k-средних для первичной сегментации изображения. На МР-снимках есть много областей с одинаковой интенсивностью серого цвета, что приводит к множеству локальных минимумов, которые увеличивают чрезмерную сегментацию, когда мы приме-

няем алгоритм водораздела. Поэтому, при первичной сегментации грубые участки изображения буду сглажены. Выбор метода k-средних обусловлен тем, что он прост в применении и имеет относительно низкую вычислительную сложность. Более того, он отлично справляется с задачей сегментации биомедицинских изображений, так как количество кластеров, на которое разбивается исходное изображение заранее определяется областями тела человека, которые на нём изображены [12]. МР-изображение головы в целом состоит из областей, представляющих кость, мягкие ткани, жир и фон. Следовательно, целесообразно выбрать $k = 4$.

Набор данных разделен на k кластеров и точки из набора случайным образом распределяются по кластерам, в результате чего кластеры имеют примерно одинаковое количество точек. Каждый кластер в наборе определен центром кластера и членом кластера. Центром можно определить точку, в которой сумма расстояний от неё до всех остальных членов кластера минимальна. Таким образом, алгоритм k-средних – это итеративный алгоритм, который минимизирует сумму расстояний от каждого объекта кластера до его центра и применяется ко всем кластерам [1].

Результат применения алгоритма k-средних изображен на рис. 1 и рис. 2.

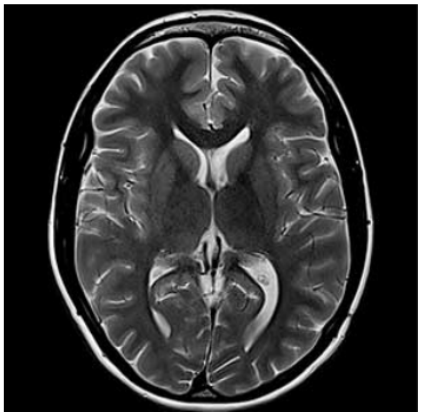


Рис. 1 Сегментируемое изображение

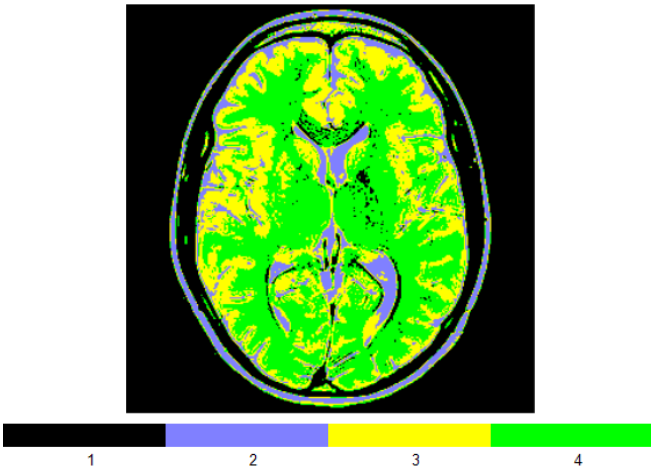


Рис. 2 Результат работы алгоритма при $k = 4$ с маркировкой

2. Модифицированный метод водораздела

Алгоритм водораздела – это метод сегментации, основанный на топологической теории и заимствовании математической морфологии. Он имеет преимущества быстрой скорости вычисления, точного позиционирования границ и ширины одного пикселя, закрытые и точные границы сегментируемой области [2]. Однако он чувствителен к детализированным текстурам и световому шуму изображения. Некоторые важные линии границ областей у изображений с низкой контрастностью нестабильны, поэтому изображение очень легко может быть сегментировано с низкой детализацией. Чрезмерная сегментация является сложной проблемой. Чтобы ее решить, многими исследователями предложены различные улучшения алгоритма, которые можно условно разделить на несколько видов. Одним из них является предварительная обработка изначального изображения при помощи градиентного изображения, удаления светового шума и маркировки целевой области перед применением метода водораздела [3–5].

В данном случае, для получения градиентного изображения, мы применяем оператор Собеля, так как он является оптимальным по вычислительной сложности и эффекту сглаживания, что как раз уменьшает световой шум изображения. Таким образом, в отличие от оригинального метода водораздела, здесь используется не оригинальное изображение, а гра-

диентное, в результате чего уменьшается количество неверных границ областей, получаемых в результате применения алгоритма. Затем алгоритм моделирования дождевых осадков применяется к улучшенной карте границ. Самый резкий «спуск» для дождевых капель реализуется с помощью сетки 3 × 3, сфокусированной на каждом участке градиентной карты. Находим локальный минимум (по интенсивности серого цвета) среди оставшихся 8 пикселей. Маркируем данный пиксель и смещаем сетку в его сторону. Процесс маркировки пикселей и смещения сетки повторяется до тех пор, пока не будет достигнут локальный минимум. Пиксели, которые составляют путь от изначально выбранного пикселя до локального минимума, получают метку этого минимума. Повторяем данные действия, отслеживая путь наискорейшего спуска для всех пикселей, которые ещё без метки. Пиксели, через который лежит путь достижения общего минимума должны использовать ту же метку, что и этот минимум, и таким образом они составляют «водосборную полость», которая относится к перегородке(водоразделу) в изображение. Эти перегородки представляют собой начальную пиксельную карту сегментации. Таким образом, получаем алгоритм состоящий из следующих шагов:

Шаг 1:

Применяем оператор Собеля к полученному изображению после первичной сегментации алгоритмом k-средних, получая таким образом градиентное изображение.

Шаг 2:

Произвольно выбираем пиксель на градиентном изображении. Он будет центром сетки 3 × 3 пикселя.

Шаг 3:

Из 8 оставшихся пикселей (без центрального) выбираем пиксель с наименьшей интенсивностью серого цвета. Помечаем его как локальный минимум в данной сетке.

Шаг 4:

Смещаем сетку в сторону пикселя с меткой локального минимума, теперь он становится центральным.

Шаг 5:

Повторяем **Шаг 3** и **Шаг 4** пока не останется пикселей с интенсивностью серого цвета меньшей, чем у центрального в данной сетке. Помечаем его как локальный минимум. Все пиксели, которые являлись центром вплоть до последнего, помечаем маркером последнего.

Шаг 6:

Произвольно выбираем новый центральный пиксель так, чтобы в сетку не попали уже промаркированные пиксели. Повторяем **Шаг 3 – Шаг 5** пока нельзя будет наложить сетку на пиксели изображения.

Шаг 7:

Оставшиеся пиксели помечаем одним маркером. Полученный результат будет являться сегментационной картой.

Первоначальная карта страдает от чрезмерной сегментации. Следовательно, мы реализуем пост-сегментационный процесс слияния в модифицированном алгоритме водораздела. Целью данного процесса является уменьшение количества разбиений, полученных в ходе сегментации, при этом не ухудшая точность и чёткость изображения.

Пусть $Im(x, y)$ – исходное изображение. Первоначальное разбиение точек, полученное при помощи метода водораздела, образует множество $R = \{R_1, ..., R_N\}$, где R_i – i-й набор и N – общее число наборов. Таким образом, алгоритм состоит из следующих шагов:

Шаг 1:

Обозначаем размер каждого из наборов как N_i.

Шаг 2:

Для каждого $i = \overline{1, N}$ вычислить среднюю интенсивность M_i наполненности набора R_i по формуле

$$M_i = \frac{1}{N_i} \sum_{(x,y)\in R_i} Im(x,y).$$

Шаг 3:

Для каждой пары наборов i и j выполнить:

Шаг 3.1: Определить разницу средних интенсивностей

$$M_{ij} = |M_i - M_j|;$$

Шаг 3.2: Определить разницу интенсивностей наборов

$$B_{ij} = \frac{1}{N_{ij}} \sum_{\{(x_i,y_i),(x_j,y_j)\}} \left| Im(x_i,y_i) - Im(x_j,y_j) \right|,$$

где $(x_i, y_i) \in R_i$ и $(x_j, y_j) \in R_j$ являются всеми 8-ю соседними пикселями, что лежат на границах наборов R_i и R_j, а число N_{ij} – число пограничных пикселей между наборами i и j.

Шаг 4:

Определим критерий C_{ij}, который будет величиной схожести величин интенсивности между наборами i и j и определяется как

$$C_{ij} = \frac{1}{2}(M_{ij} + B_{ij}).$$

После определения C_{ij} для всех наборов, пользователь определяет пороговое значение, которому должен удовлетворять критерий перед объединением наборов. Если значение критерия меньше, то это означает, что наборы i и j являются одинаковыми на основе пространственных критериев, а значит их следует объединить.

3. Результаты тестирования

Для оценки работоспособности представленных выше алгоритмов кластеризации, была реализована программа, включающие в себя модули на языках C++, C#, Python с использование библиотеки искусственного зрения OpenCV. Целью эксперимента является сравнение результатов обработки классическим методом водораздела и предложенной методологией. Тестовыми данными были выбраны 10 снимков МРТ головного мозга. В 8 из 10 случаев, результат сегментации изображений сильно отличался: предложенная методика объединяла около 90 % изначальных разбиений, в то время как классический метод лишь 30 %. В остальных двух случаях, процент объединения составлял около 45 %. Данный факт можно отчётливо увидеть на рис. 3–6.

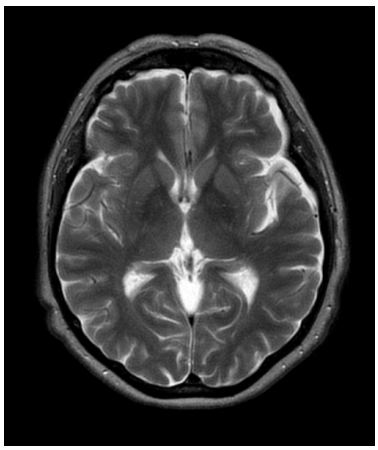


Рис. 3. Исходное изображение

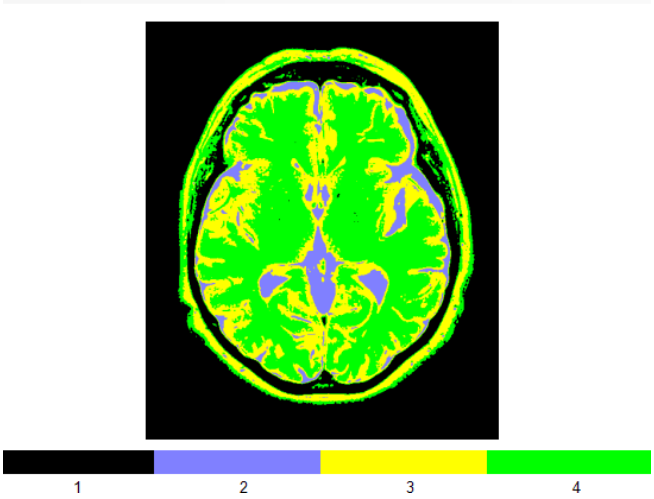


Рис. 4. Результат алгоритма k-средних при $k = 4$

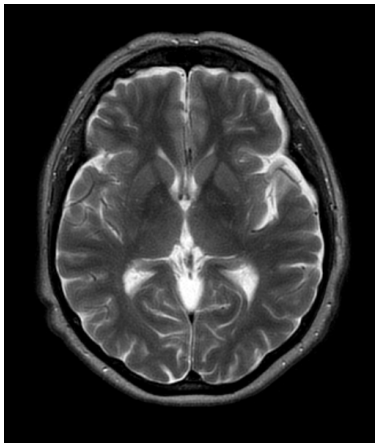


Рис. 5. Результат классического метода водораздела

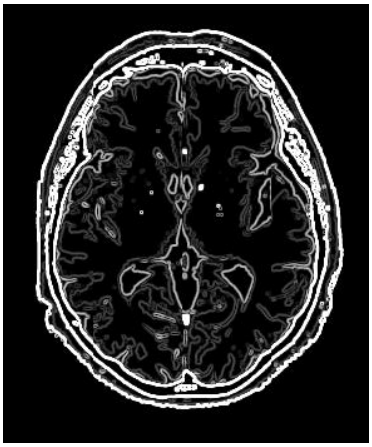


Рис. 6. Результат предложенной модификации

В частности, рассматривая рис. 5 и рис. 6, можно чётко увидеть сильную разницу в количестве полученных областей. Количественно же, обработка классическим методом водораздела дала 2756 областей, в то время как предложенный вариант лишь 348. Таким образом можно сделать вывод о том, что предложенные в статье модификации более эффективны, нежели классический метод.

Заключение

В данной статье были рассмотрены способы обработки медицинских МР изображений и была предложена модификация одного из таких способов. Она устраняет недостатки классического метода водораздела, в частности возможность чрезмерной сегментации и чувствительность к световому шуму. Результат эксперимента показал, что модификация является эффективной и её использование даёт более чёткое изображение, которое будет более эффективно использовано экспертом в медицине.

Литература

1. *Черняев, Д. Е.* Обзор алгоритмов кластеризации для сегментации изображений / Д. Е. Черняев // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 7–9 декабря 2020 г. – Воронеж, 2021. – С. 470–471.

2. *Pham, D. L.* Adaptive fuzzy segmentation of magnetic resonance images / D. L. Pham, J. L. Prince // Transactions on Medical Imaging. – 1999. – V. 18. – P. 737–752.

3. *Masood, S.* A survey on Medical Image segmentation / S. Masood, M. Sharif, A. Masood, M. Yasmin, M. Razar //Current Medical Imaging Reviews. – 2015. – V. 11. – P. 3–14.

4. Pham, D. L. An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneties / D. L. Pham, J. L Prince //Pattern Recognition Letters. – 1999. – V. 20. – P. 57–68.

5. *Grau, V.* Improved watershed transform for medical image segmentation using prior information / V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, S.K. Warfield // Transactions on Medical Imaging. – 2004. – V. 23. – P. 447–458.

6. *Bezdek, J. C.* Review of MR image segmentation techniques using pattern recognition // J. C. Bezdek, L. O. Hall, L. P. Clarke // Medical Physics. – 1993. – V. 20. – P. 1033–1048.

7. *Marr, D.* Vision. – New York : Freeman and Company, 1982.

8. *Jain, A. K.* Algorithms for Clustering Data / A. Jain, R. Dubes // Prentice Hall. – 1988.

9. *Haiyang, L.* Dynamic particle swarm optimization and k-means clustering algorithm for image segmentation / L. Haiyang, H. Hongzhou, W. Yongge // Optik. – 2015. – P. 126, 4817– 4822.

10. *Halder, A.* Dynamic Image segmentation using Fuzzy c-means based Genetic Algorithm / A. Halder , S. Pramanik, A. Kar //International Journal of Computer Application. – 2011. – V. 28. – P. 12–18.

11. *Bezdek, J. C.* Pattern Recognition with Fuzzy Objective Function Algorithms / J. C. Bezdek, C. James // Medical Physics. – 1981.

12. *Chen, C. W.* Image segmentation via adaptive K-mean clustering and knowledge based morphological operations with biomedical applications / C. W. Chen, J. Luo, K. J. Parker // Transactions on Image Processing. – 1998. – V. 7. – P. 1673–1683.

Черняев Дмитрий Евгеньевич – студент 4-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: dcherhtc@gmail.com

Леденева Татьяна Михайловна (научный руководитель) – д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: ledeneva-tm@yandex.ru

УДК 004.942

ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ И ЕГО ПРИМЕНЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ АГЕНТОВ

А. Г. Шенгелия

Воронежский государственный университет

Введение

Социальное моделирование – это область исследований, которая применяет вычислительные методы для изучения проблем в социальных науках. Исследуемые вопросы включают в себя проблемы вычислительного права, психологии, организационного поведения, социологии, политологии, экономики, антропологии, географии, инженерии, археологии и лингвистики [1].

Социальное моделирование направлено на преодоление разрыва между описательным методом, используемым в социальных науках, и формальным методом, используемым в естественных науках, путем смещения акцента на процессы/механизмы/поведение, которые строят социальную реальность. В социальном моделировании компьютеры поддерживают человеческую мыслительную деятельность, используя эти механизмы. Эта область исследует моделирование обществ как сложных нелинейных систем, которые трудно изучать с помощью классических математических моделей, основанных на уравнениях.

Агентное моделирование – это система, в которой агенты взаимодействуют в совокупности независимо друг от друга.

1. Моделирование социума

1.1. Моделирование на системном уровне

Моделирование на системном уровне – это самый старый уровень социального моделирования. Моделирование на системном уровне рассматривает ситуацию в целом. Этот теоретический взгляд на социальные ситуации использует широкий спектр информации для определения того, что должно произойти с обществом и его членами при наличии определенных переменных, какую реакцию они должны иметь на новую ситуацию. Навигация по этому теоретическому моделированию позволит исследователям разработать обоснованные идеи о том, что произойдет при некоторых конкретных переменных. Например, если бы НАСА проводило моделирование на системном уровне, это принесло бы пользу организации, предоставив экономически эффективный метод исследования для навигации по моделированию.

1.2. Агент-ориентированное социальное моделирование

Агент-ориентированное социальное моделирование состоит из моделирования различных обществ и размещения искусственных агентов в компьютерно-моделируемом обществе для наблюдения за их поведением. Из этих данных можно узнать о реакциях искусственных агентов и перевести их в результаты реальных агентов и симуляций. Три основные области – агентные вычисления, социальные науки и компьютерное моделирование. Именно здесь разрабатываются и теоретизируются социальные явления. Основной целью агент-ориентированного социального моделирования является предоставление моделей и инструментов для агентного моделирования социальных явлений. С его помощью мы можем исследовать различные исходы для явлений, которые мы, возможно, не сможем увидеть в реальной жизни, и получить ценную информацию о результатах социальных явлений.

1.3. Агентное моделирование

Агентное моделирование – это экспериментальный инструмент для теоретических исследований. Оно позволяет иметь дело с более сложными индивидуальными формами поведения, такими как адаптация. В целом, с помощью этого типа моделирования создатель, или исследователь, стремится моделировать поведение агентов и связь между ними, чтобы лучше понять, как эти индивидуальные взаимодействия влияют на всю популяцию. По сути, это способ моделирования и понимания различных глобальных закономерностей.

Каждый отдельный агент отвечает за различные формы поведения, которые приводят к коллективному поведению. Эти модели поведения в целом помогают определить работу сети. Оно фокусируется на человеческих социальных взаимодействиях и на том, как люди работают вместе и общаются друг с другом, не имея единого группового разума. Это, по существу, означает, что он имеет тенденцию сосредотачиваться на последствиях взаимодействий между людьми (агентами) в популяции. Исследователи могут лучше понять этот тип моделирования, используя динамику на меньшем, более локализованном уровне. Простые индивидуальные правила или действия могут привести к согласованному групповому поведению. Изменения в этих индивидуальных действиях могут повлиять на группу в любой выбранной популяции.

2. Обучение с подкреплением

Обучение с подкреплением – область машинного обучения, целью которой является обучение агента для оптимального решения конкретных задач в конкретной среде [2].

В отличие от дискриминативного и генеративного моделирования, направленных на минимизацию функции потерь по набору наблюдений, обучение с подкреплением стремится максимизировать долгосрочное вознаграждение агента в данном окружении. Этот подход часто описывается как одна из трех основных ветвей машинного обучения наряду с обучением с учителем (прогнозирование с использованием маркированных данных) и обучением без учителя (обучение по немаркированным данным). Для начала познакомимся с некоторыми ключевыми терминами, относящимися к обучению с подкреплением:

Окружение (среда) – мир, в котором действует агент. Характеристики мира определяются набором правил, которые управляют процессом обновления состояния игры и распределением вознаграждений, с учетом предыдущих действий агента и текущего состояния игры.

Агент – сущность, предпринимающая действия в окружающей, реально складывающейся или смоделированной обстановке.

Действие – ход, который может сделать агент.

Вознаграждение – ценность, возвращаемая агенту средой после выполнения действия. Агент стремится максимизировать долгосрочную сумму своих вознаграждений [2].

Обучение с подкреплением – один из способов машинного обучения, в ходе которого испытуемая система (агент) обучается, взаимодействуя с некоторой средой. С точки зрения кибернетики, является одним из видов кибернетического эксперимента. Откликом среды (а не специальной системы управления подкреплением, как это происходит в обучении с учителем) на принятые решения являются сигналы подкрепления, поэтому такое обучение является частным случаем обучения с учителем, но учителем является среда или её модель. Также нужно иметь в виду, что некоторые правила подкрепления базируются на неявных учителях, например, в случае искусственной нейронной среды, на одновременной активности формальных нейронов, из-за чего их можно отнести к обучению без учителя.

Агент воздействует на среду, а среда воздействует на агента. О такой системе говорят, что она имеет обратную связь. Такую систему нужно рассматривать как единое целое, и поэтому линия раздела между средой и агентом достаточно условна. Конечно, с анатомической или фи-

зической точек зрения между средой и агентом (организмом) существует вполне определённая граница, но если эту систему рассматривать с функциональной точки зрения, то разделение становится нечётким. Например, резец в руке скульптора можно считать либо частью сложного биофизического механизма, придающего форму куску мрамора, либо частью материала, которым пытается управлять нервная система [3].

Заключение

Как было показано, обучение с подкреплением это современный и эффективный способ проводить моделирование социального поведения агентов в моделируемых условиях.

Ключевым компонентом создаваемой архитектуры является генеративная модель, способная вычислить распределение вероятностей для следующего возможного состояния с учетом текущего состояния и предполагаемого действия. Получив представление о физике окружающей среды с помощью случайных движений, модель может затем научиться решать совершенно новые задачи, действуя исключительно в рамках своего внутреннего представления об окружающей среде.

Литература

1. *Hughes H. P. N., Clegg C. W., Robinson M. A., Crowder R. M.* Agent-based modelling and simulation: The potential contribution to organizational psychology // Journal of Occupational and Organizational Psychology. – 2012. – 85 (3). – 487–502.

2. *Фостер Д.* Генеративное глубокое обучение. Творческий потенциал нейронных сетей. – СПб. : Питер, 2020. – 336 с.

3. *Гельфанд И. М., Пятецкий-Шапиро И. И., Цетлин М. Л.* О некоторых классах игр и игр автоматов // Докл. АН СССР, 1963. – Т. 152, № 4. – С. 845–848.

Шенгелия Артем Геннадьевич – студент 2-го курса магистратуры кафедры ERP-систем и бизнес процессов Воронежского государственного университета.
E-mail: Artem-shengeliya@ yandex.ru

Сафронов Виталий Владимирович (научный руководитель) – канд. техн. наук, доц., доцент кафедры кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: vitolik@bk.ru

УДК 004.9

АВТОМАТИЗАЦИЯ ДЕЯТЕЛЬНОСТИ ФИТНЕС-КЛУБА

А. А. Щеглеватых

Воронежский государственный университет

Введение

Для каждого человека всегда было и остается важным сохранить свое здоровье. Одним из способов достижения данной цели является регулярные занятия спортом. В современном мире понятие «физкультура» часто заменяется более броским термином «фитнес».

В городах растет количество фитнес-клубов: как филиалов крупных федеральных сетей, так и маленьких частных студий или фитнес школ. Перед каждым клубом возникает задача учета посетителей, тренировок и т. д. Вести бумажный учет в наше время не только «не современно», но и очень трудоемко, а также влечет большое количество ошибок.

Крупные федеральные сети выделяют значительные средства на разработку, сопровождение и дальнейшую поддержку единых для всей сети автоматизированных систем учета. Небольшим залам и студиям, зачастую, дешевле и проще разработать и внедрить свое решение данной задачи. В нем будет учтена специфика работы данного заведения, при отсутствии неиспользуемых функций, которыми буквально пестрят универсальные решения. В свою очередь такое решение отличается быстрым внедрением, простотой обучения персонала и дальнейшего использования.

В данной работе описывается процесс проектирования и разработки web-приложения для автоматизации работы фитнес-клуба.

Целью работы является разработка приложения с удобным пользовательским интерфейсом для работы с базой данных фитнес-клуба. Разработанное приложение предназначено для администраторов, тренеров и клиентов клуба.

1. Анализ

1.1. Общий анализ

Для решения поставленной задачи необходимо разработать пакет инструментов, позволяющий выполнять действия: регистрировать нового пользователя в системе, добавлять информацию о новых тренировках и тренерах, записывать клиента на персональные тренировки к тренеру, осуществлять e-mail – рассылку, отображать статистику тренировок и т. д. Для удобства взаимодействия пользователя и приложения понадобятся вспомогательные библиотеки.

Общий вид взаимодействия пользователя с приложением представлен на рис. 1.1.

1.2. Анализ существующих решений

В настоящее время существует большое количество как локальных, так и Web-приложений для решения поставленной задачи. В табл. 1 представлены 3 популярных приложения.

Рис. 1.1. Диаграмма вариантов использования

Таблица 1

Сравнительная характеристика существующих систем

Название	Тип	Описание
«1С: Фитнес клуб»	Локальное/web	Доступно на всех платформах. Существует как локальная версия приложения, так и облачная. Многофункциональное приложение, визуально приятное для восприятия, однако интерфейс рассчитан на хорошо подготовленного пользователя. Кроме того, маленькие организации могут не нуждаться в большей части функций, тогда предприятие вынуждено переплачивать за те услуги, которыми оно не пользуется.
«Dance Studio»	Локальное	Доступно только на платформе Windows. Интуитивно понятный интерфейс, выполненный в спокойной цветовой гамме. Такое приложение не требует особых навыков владения ПК, что делает его удобным в использовании. На официальном сайте есть обучающие видео для персонала.
«Lucky Fit»	Web	Облачное приложение, которое доступно на всех платформах. Как и многие «фитнес-приложения», выполнено в бело-синих цветах, имеет весь необходимый функционал даже в бесплатной версии. Однако в нем нет ни сетки расписания, ни возможности узнать какую-либо информацию о персональных или групповых тренировках. Ориентировано больше на менеджеров.

2. Реализация

2.1. Инструменты реализации

Для разработки базы данных был выбран декларативный язык SQL-структурированный язык запросов, который дает возможность создавать и работать в реляционных базах данных, являющихся наборами связанной информации, сохраняемой в таблицах. Работа с ним производилась по средствам СУБД «MySQL», т.к. она имеет большой функционал по работе с базой данных, высокую скорость работы базы данных и является безопасной.

Решение поставленной задачи реализовано по средствам языка JavaScript, т.к. данный язык позволяет разрабатывать не только клиентскую часть приложения, но и серверную. Для разработки пользовательского интерфейса была выбрана библиотека React.js, а для выполнения кода на стороне сервера програмная платформа Node.js

2.2. Структура данных

Проект имеет трехзвенную архитектуру, представленную на рис. 2.1.

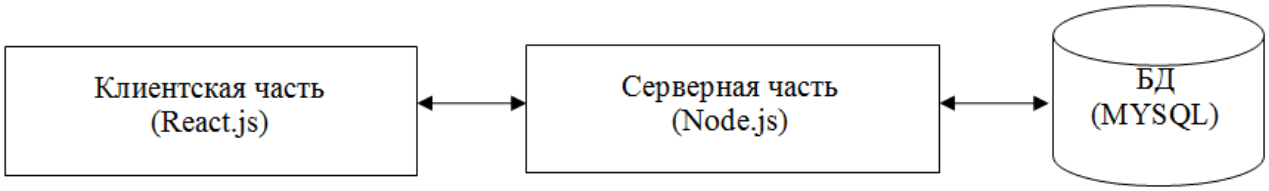


Рис. 2.1.Схема архитектуры приложения

Для работы с базой данных был использован MySQL Community Server 8.0.23 и инструмент для визуального проектирования баз данных MySQL Workbench. Разработанная база данных должна хранить всю необходимую информацию для обеспечения корректной работы приложения. Схема базы данных представлена на рис. 2.2.

Рис. 2.2.Схема базы данных

Серверная часть выполнена при помощи Node.js и фреймворка Express. Она отвечает за подключение к базе данных, подключение и настройку почтового клиента Nodemailer, логику регистрации и авторизации пользователя в системе, обработку запросов к базе данных и отправку результатов запросов клиентской части.

Клиентская часть реализована при помощи библиотеки React.js и фреймворка Material-UI. React позволяет сделать web-страницы живыми, а Material-UI придает опрятный и «современный» внешний вид сайту. Для большей информативности на сайте представлены изображения тренировок, фотографии клиентов и тренеров. Для того чтобы не нагружать БД полноценными изображениями, было принято решение хранить все необходимые медиа-файлы в приложении (client/public/img), а в БД заносить только название изображения и получать его через путь. Данная часть приложения ответственна за отрисовку компонент на странице, работу с данными, полученными от сервера, маршрутизацию страниц и отображения общих частей приложения, таких как: футер, меню и логотипа загрузки.

Структура приложения отображена на рис. 2.3.

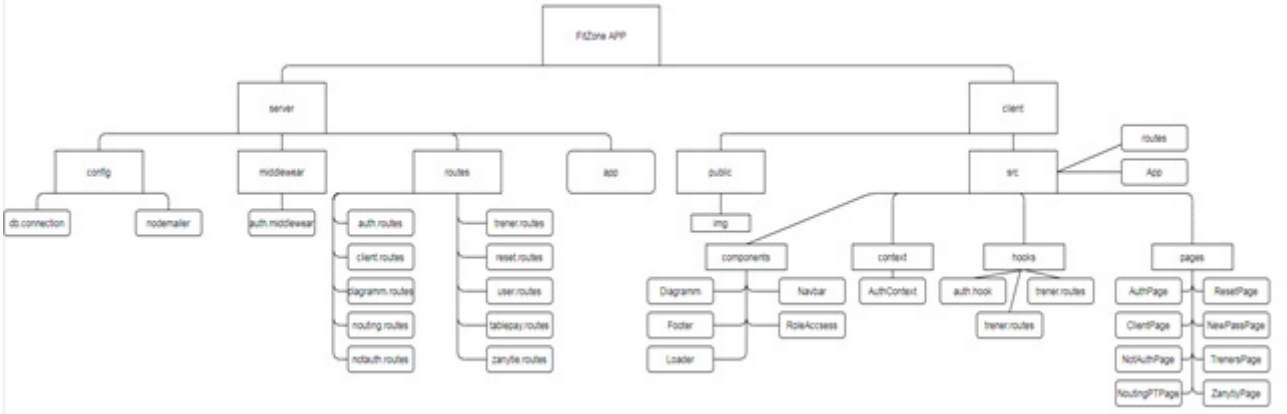


Рис. 2.3.Схема приложения

2.3. Маршрутизация

Для корректного отображения web-страниц необходимо использовать компонент Route библиотеки react-router-dom.

Под маршрутизацию в проекте выделен отдельный файл /client/src/routes, в нем определены все маршруты страниц для зарегистрированных и незарегистрированных пользователей. Для определения маршрута используется компонент Switch библиотеки react-router-dom. Если отображение информации на странице зависит от конкретных данных пользователей, то используется контекст: <AuthContext.Consumer>.

Заключение

В результате проделанной работы было реализовано и протестировано web-приложение для автоматизации работы фитнес-клуба, функциональные возможности которого включают:

- регистрацию в системе;
- авторизацию в системе;
- восстановление пароля по адресу электронной почты;
- просмотр информации о групповых/персональных тренировках;
- просмотр информации о тренерах;
- просмотр персональной информации;
- запись на персональные/групповые тренировки;
- подтверждение/отмена записи на персональную тренировку;
- редактирование информации о тренерах и групповых тренировках;
- просмотр статистики тренеров и тренировок для администратора;
- общение с чат-ботом для незарегистрированных пользователей.

На данный момент обсуждаются детали внедрения данного web-приложения в работу Федерации ушу г. Липецка и регистрации приложения на хостинге.

В дальнейшем работа над приложением будет продолжена, планируется добавить оплату абонементов и персональных тренировок при помощи сервиса «ЮMoney».

Литература

1. Tutorials and references relating to HTML, CSS, JavaScript, JSON. – Режим доступа: https://www.w3schools.com

2. Онлайн учебник javascript. – Режим доступа: https://learn.javascript.ru

3. Material-UI: A popular React UI framework. – Режим доступа: https://material-ui.com

4. React – JavaScript library for building user interfaces. – Режим доступа: https://reactjs.org/

5. Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. – Режим доступа: https://nodejs.org/en

Щеглеватых Алина Александровна – студентка 4-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: alinka030499@mail.com

Борисенков Дмитрий Васильевич (научный руководитель) – канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: xuser@relex.ru

Содержание

Научное издание

Математика, информационные технологии, приложения

Сборник трудов
Межвузовской научной конференции
молодых ученых и студентов

Воронеж,
26 апреля 2021 г.

Подписано в печать 31.05.2021. Формат 60 × 84 / 8.
Усл. печ. л. 32,2. Тираж 100 экз. Заказ № 105.

Отпечатано с готового оригинал-макета

ООО Издательско-полиграфический центр
«Научная книга»
394030, г. Воронеж, ул. Никитинская, д. 38, оф. 308
Тел. +7 (473) 200-81-02, 200-81-04
http://www.n-kniga.ru; e-mail: zakaz@n-kniga.ru

Отпечатано в типографии ООО ИПЦ «Научная книга».
394026, г. Воронеж, Московский пр-т, 11б
Тел. +7 (473) 220-57-15, 238-02-38
http://www.n-kniga.ru; e-mail: typ@n-kniga.ru