

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

**СБОРНИК СТУДЕНЧЕСКИХ
НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ
НАУК ВГУ**

В ы п у с к 14

Часть 1

Научные работы студентов бакалавриата

Под редакцией кандидата технических наук Д.Н. Борисова

Воронеж
Издательский дом ВГУ
2020

УДК 004.65 + 004.438.5
ББК Ч481(2)22
С23

Рекомендовано к печати
Ученым советом факультета компьютерных наук ВГУ

Сборник студенческих научных работ факультета компьютерных наук ВГУ. Выпуск 14 : в 2 ч. / под ред. Д. Н. Борисова ; Воронежский государственный университет. – Воронеж : Издательский дом ВГУ, 2020.
ISBN 978-5-9273-3058-4

Ч.1. : Научные работы студентов бакалавриата. – 209 с.
ISBN 978-5-9273-3059-1

В сборник включены научные работы студентов бакалавриата факультета компьютерных наук ВГУ, выполненные в 2019–2020 гг. под руководством преподавателей факультета, представленные в виде докладов и рекомендованные к опубликованию оргкомитетами студенческих научных конференций.

УДК 004.65 + 004.438.5
ББК Ч481(2)22

ISBN 978-5-9273-3059-1 (ч. 1)
ISBN 978-5-9273-3058-4

© Воронежский государственный университет, 2020
© Оформление. Издательский дом ВГУ, 2020

Визуальное фенотипирование растений с использованием алгоритмов компьютерного зрения

Т. В. Акиндинова

Студент бакалавриата

М. А. Дрюченко

Доцент

Введение

Большинство информации об окружающем мире мы получаем при помощи зрения, поэтому в научных исследованиях появилась перспектива снабжения технических средств возможностью «видеть». Для этого создаются целые программные комплексы, использующие различные методы обработки, анализа и распознавания изображений.

Данная работа посвящена применению алгоритмов компьютерного зрения для решения задачи фенотипирования кустов томатов. Тема является актуальной, так как программные комплексы, использующие алгоритмы машинного зрения, позволяют дать точную оценку параметров растений в удобной для человека форме и за небольшое время, что значительно повышает производительность хозяйственных предприятий. Общая задача фенотипирования была разделена на 4 этапа, которые описаны ниже.

1. Формирование панорамных изображений

Исходными данными являются изображения частей кустов томатных культур, полученные при помощи матричных или линейных камер. В общем случае существует два типа алгоритмов формирования панорамных изображений: дескрипторные и корреляционные. Дескрипторные алгоритмы работают с ключевыми точками изображения, которые содержат важную информацию об объекте. Далее формируются векторы признаков, обеспечивающих инвариантность к нахождению одних и тех же точек на разных изображениях. К данной группе алгоритмов относится алгоритм SIFT, результаты работы которого показаны на рис. 1 и рис. 2.

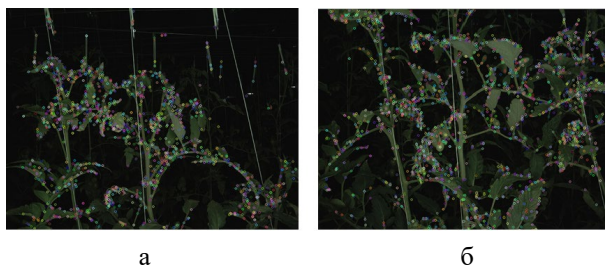


Рис. 1. Пример работы алгоритма SIFT (найденные ключевые точки): а – верхняя часть куста, б – нижняя соседняя часть

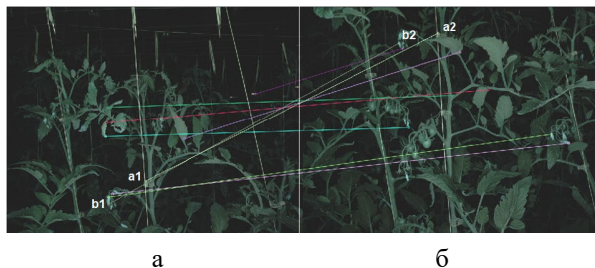


Рис. 2. Пример работы алгоритма сопоставления ключевых точек по паре соседних кадров: а – верхняя часть куста, б – нижняя соседняя часть

Дескрипторные алгоритмы не дают хороших результатов из-за сложной формы объекта, различной освещённости его частей и большого шага при отсеме. В итоге находится достаточное количество ключевых точек, которые в действительности не содержат важную информацию об объекте, также при их сопоставлении часто возникают ошибки.

Идея корреляционных алгоритмов заключается в попиксельном сравнении изображении [1]. Сначала выбирается мера сходства, в самом простом случае используется функция взаимного смещения $(\Delta x, \Delta y)$, которая может быть вычислена как квадрат СКО яркостей соответствующих пикселей:

$$\sum_{f_1, f_2} (\Delta x, \Delta y) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (f_1(x, y) - f_2(x - \Delta x, y - \Delta y))^2, \quad (1)$$

где f_1 и f_2 – изображения одного и того же объекта, отличающиеся только сдвигом; $N \times N$ – размер области, в которой вычисляется

корреляция. Далее происходит минимизация выбранного критерия или максимизация корреляции. Для работы этих алгоритмов можно использовать информацию о расположении белых нитей, к которым прикрепляются растения. Результат работы корреляционного сопоставления представлен на рис. 3.

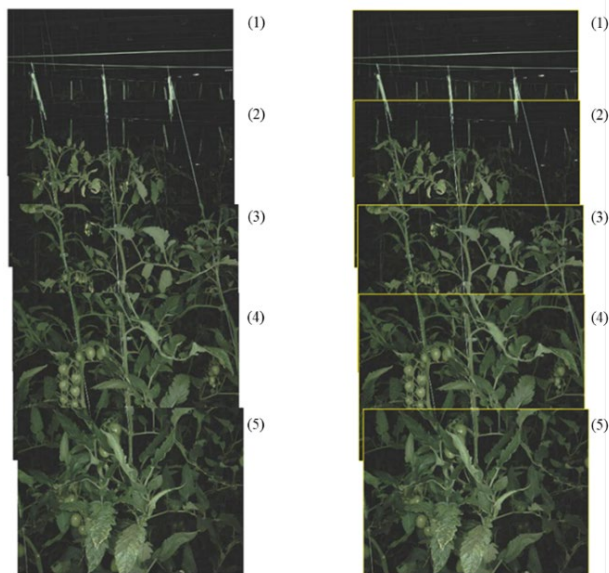


Рис. 3. Пример работы алгоритма корреляционного сопоставления с использованием дополнительной информации о расположении нитей

Однако такое сопоставление также не является надёжным, так как далеко не на всех изображениях нити отчётливо видны.

Для корректного решения задачи растровой сшивки можно воспользоваться информацией о физических данных аппаратного комплекса. Камеры располагаются на некотором расстоянии S от кустов растений, двигаются с фиксированным шагом и имеют фокусное расстояние F . Также у них есть определённый размер матрицы по горизонтали h и вертикали v . Чтобы рассчитать реальный размер объекта $H \times V$ необходимо воспользоваться следующими формулами:

$$H = \frac{hS}{F}, \quad (2)$$

$$V = \frac{vS}{F}. \quad (3)$$

Используя эту информацию, можно рассчитать значения наложений между соседними кадрами и выполнить растровую сшивку соседних кадров.

2. Отделение полезных объектов от фона изображений

Для расчёта биологических параметров необходима информация лишь о растениях переднего плана, которые необходимо отделить от сложного фона. Для решения этой задачи я применяла три различных алгоритма. Первый алгоритм основывается на использовании сверточных нейронных сетей (НС) и прямоугольных фрагментов в качестве обучающих данных. Его достоинствами является относительная простота подготовки обучающей выборки и полное покрытие кадров за счёт непересекающихся фрагментов. Недостатками алгоритма является низкая точность распознавания вдоль сложных по форме границ объектов и требование проведения ручной разметки.

Для решения задачи сегментации тепличных культур томатов будут использоваться типовые архитектуры НС, построенных из сверточных и субдискретизационных слоёв с последующими функциями активации. Также может использоваться механизм прореживания нейронов, которые не будут в дальнейшем влиять на процесс обучения сети. После построения сверточной нейронной сети нужно её обучить. Для этого на изображениях, размеченных по сетке с фиксируемым шагом, проводится ручная разметка, путём нажатия левой и правой кнопок мыши. Процесс формирования обучающих данных представлен на рис. 4.

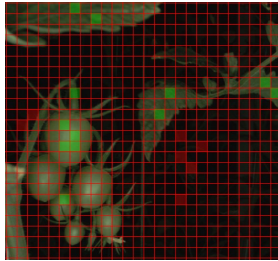


Рис. 4. Ручная разметка растровых фрагментов по классам

После обучения сети её тестируют на растровых изображениях, которые так же размечаются по сетке и подаются на вход нейронной сети.

Второй алгоритм основан на суперпиксельной сегментации и сверточных нейронных сетях. Его достоинством является высокая точность распознавания вдоль границ объектов, а недостатками – относительно большое время работы и сложность в формировании обучающих данных. В своей работе я использовала алгоритм SLIC [2], который является модификацией k -средних, только области поиска ограничиваются некоторыми окрестностями. В качестве меры близости используется взвешенная сумма евклидовых расстояний по координатам и трём цветовым компонентам

$$D = \sqrt{d_c^2 + \left(\frac{d_s^2}{S}\right)m^2}, \quad (4)$$

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}, \quad (5)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \quad (6)$$

где d_s – пространственное расстояние, определяемое координатами пикселей, d_c – цветовое расстояние, D – сумма цветового расстояния и нормализованного пространственного расстояния по плоскости XU через интервал сетки S .

После получения суперпикселей на изображениях происходит ручная разметка по классам для формирования обучающей выборки. Затем приведение суперпикселей к прямоугольной форме с помощью алгоритмов интерполяции. Данный процесс изображен на рис. 5.

Сформированные прямоугольные фрагменты можно использовать как обучающие данные или подавать в таком виде на вход обученного нейронного классификатора для распознавания

Третий алгоритм основывается на применении сверточных сетей архитектуры Unet и размеченным по классам изображениям. Достоинством данного подхода является высокая точность распознавания вдоль границ объектов, а недостатком – сложность в формировании данных для обучения. Unet отличается от типовых сверточных архитектур тем, что на выходе она имеет изображение с цветовой разметкой по классам того размера, что и входное растровое изображение. В нашем случае это будут бинарные изображения, содержащее в себе два класса: объект и фон.

Для получения эталонных бинарных изображений предполагается проведение черновой бинаризации с порогом [3], значение которого выбирается при помощи гистограммы яркости пикселей изображений.

Затем происходит поиск и заполнение контуров малых площадей внутренних частей растений.

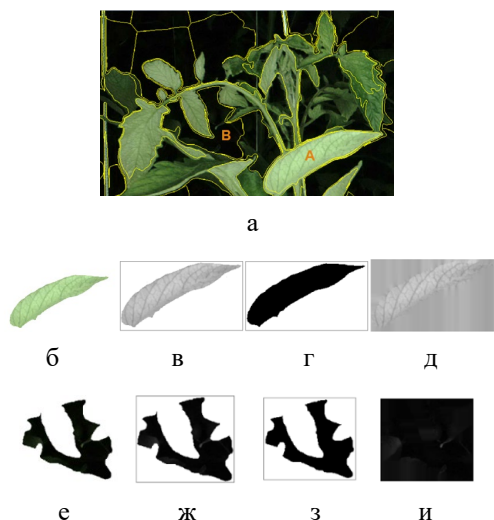


Рис. 5. Работа алгоритма: а – пример работы алгоритма SLIC; б, е – исходные суперпиксели объекта и фона; в, ж – вписанные в градациях серого суперпиксели объекта и фона; г, з – бинарные маски объекта и фона; д, и – результат работы интерполяционного алгоритма для суперпикселей объекта и фона

И в завершении проводится поиск и удаление внешних контуров большей площади, принадлежащих ложно распознанным объектам. Пример работы данного алгоритма представлен на рис. 6.



Рис. 6. Результат работы алгоритма получения эталонных изображений

3. Декомпозиция распознанных объектов

На данном этапе происходит разделение объекта переднего плана на составные классы листьев, стеблей, плодов. С помощью морфологии можно найти на изображениях стебли, задав структурообразующий элемент достаточно вытянутым по оси Y и сжатым вдоль оси X . Операция предельной эрозии оставит лишь те маркеры объектов, которые будут относиться к стеблям растений. Для распознавания листьев и плодов можно использовать алгоритмы скелетизации. Скелет плода будет выглядеть как маленький прямой отрезок, а скелет листьев будет состоять из множества отростков разной длины. Результат декомпозиции объекта представлен на рис. 7.



Рис. 7. Результат декомпозиции распознанного объекта на классы

4. Оценка биометрических параметров растений

Биологические параметры растений считаются отдельно для каждого класса. Такими признаками могут быть высота стебля, площадь листьев, количество плодов и тд. Конкретные значения можно получить, используя декомпозированную разметку по классам. Например, подсчитать площадь объектов какого-либо класса можно по формуле ниже:

$$\sum_{i=1}^{W \times H} p_i, \quad (7)$$

где p_i – пиксель, значение которого равно 1, если принадлежит бинарному классу стеблей, 0 – если не принадлежит; $W \times H$ – размеры изображения.

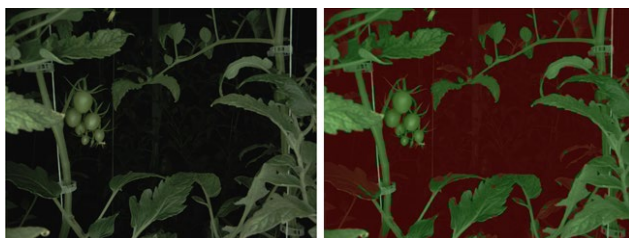
5. Тестирование разработанного программного модуля

Для оценки точности распознавания была разработана программа, реализующая рассмотренные выше алгоритмы. Формула для вероятности ошибки распознавания реальных объектов изображения I имеет следующий вид:

$$\alpha(I) = \frac{S(I_V \oplus I_B)}{\max(S(I_V), S(I_B))} \quad (8)$$

где I_B – результат сегментации изображения I одним из тестируемых алгоритмов, I_V – корректный результат сегментации, точно передающий пространственные характеристики растительных объектов, S – совокупная площадь областей принадлежащих объектам на сегментированном изображении.

Результат работы алгоритма, основанного на использовании сверточных нейронных сетей и прямоугольных фрагментов в качестве обучающих данных представлен на рис. 8.



а

б

Рис. 8. Пример работы сегментации на основе сверточных нейронных сетей и прямоугольных фрагментов: а – изображение до сегментации, б – изображение после сегментации

Заключение

В данной работе рассмотрены возможности применения алгоритмов компьютерного зрения для решения задачи визуального фенотипирования кустов томатов. Также была разработана программная реализация, позволяющая получить численные значения ошибок распознавания. При тестировании алгоритма с использованием сверточных сетей и прямоугольных растровых фрагментов средняя ошибка распознавания не превышает 8-10%, что делает алгоритм пригодным для проведения корректной сегментации полезного объекта от фона.

Список литературы

1. Потапов, А. С. Системы компьютерного зрения : учебное пособие / А. С. Потапов. – СПб: Университет ИТМО – 2016. – 161 с.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
3. Саввин, С. В. Методы суперпиксельной сегментации и их применение для анализа изображений с разнородной текстурой / С. В. Саввин, А. А. Сирота // Вестник ВГУ, Серия: системный анализ и информационные технологии – 2016. – № 4. – С. 165-173.

Разработка программного модуля для интеллектуального анализа текстовых данных новостных интернет-ресурсов

А. С. Бобраков

Студент бакалавриата

В. В. Гаршина

Доцент

Введение

В современном мире производится огромное количество интернет ресурсов с текстовой информацией. Различные новостные сайты, социальные сети и многие другие источники могут содержать множество данных для дальнейшего анализа и использования. Однако данные из таких источников отчасти представляют из себя тексты. К тому же, часть текстов, как, например, комментарии или отзывы, несет в себе оценочную информацию. В связи с этим, обработка таких ресурсов вручную становится весьма неэффективна в силу того, что человеку требуется затратить на это много времени.

Исходя из этого, можно считать, что системы, способные осуществлять мониторинг и анализ таких источников, могут быть очень полезными. Так, например, они могут применяться в государственном секторе для отслеживания социально-экономической ситуации в регионах, что позволит в определенные моменты скорректировать свою работу, с целью улучшения показателей в тех или иных отраслях.

Данная статья посвящена созданию системы, способной собирать текстовую информацию из новостных интернет-источников и

производить ее обработку, для дальнейшего анализа и использования полученных данных на практике.

В этап обработки будут входить:

- поиск именованных сущностей из текста новости;
- анализ комментариев к новости на предмет их эмоциональной окраски
- классифицирование тестов новостей по заранее созданному списку категорий

1. Поиск именованных сущностей

Поиск именованных сущностей предполагает нахождение в тексте определенных слов или группы слов, обозначающих название того или иного объекта, их классификацию и присвоение определенного тега, показывающего принадлежность к тому или иному классу.

В рамках разрабатываемого модуля производился поиск одних из основных сущностей:

- имена и фамилии людей (PER);
- названия географических объектов (LOC);
- название организации – физического или юридического лица (ORG);

Слова, не отнесенные ни к какому из перечисленных классов, будут помечаться тегом O (рис. 1).

На сегодняшний день существует ряд подходов к решению этой задачи. К ним относятся [1]:

- подходы, основанные на правилах;
- подходы, основанные на словарях;
- машинное обучение с учителем;
- машинное обучение без учителя.

Одним из передовых и часто используемым подходом является применение трансферного обучения – а именно тонкой настройки (finetuning) [2]. Это метод глубокого машинного обучения, который предполагает использование для решения поставленной задачи уже обученную сеть для решения схожей задачи, путем переучивания или изменения архитектуры последних слоев сети. Таким образом, для обучения новой сети потребуется намного меньше времени и меньший объем данных.

Именно этот подход и был выбран для реализации данной функциональности модуля.

В качестве подготовленной модели была выбрана модель BERT, обученная на корпусах русскоязычных текстов Wikipedia и корпусах новостей. Архитектурно она представляет из себя немного модифицированную часть архитектуры Transformer – 12 слоев Encoder.

Каждый такой слой представляет из себя слой внутреннего внимания (общее число этих слоев – 12) и сеть прямого распространения, имеющая 768 скрытых нейронов. К ней добавляется сети прямого распространения, последний слой которой имеет функцию активации softmax.

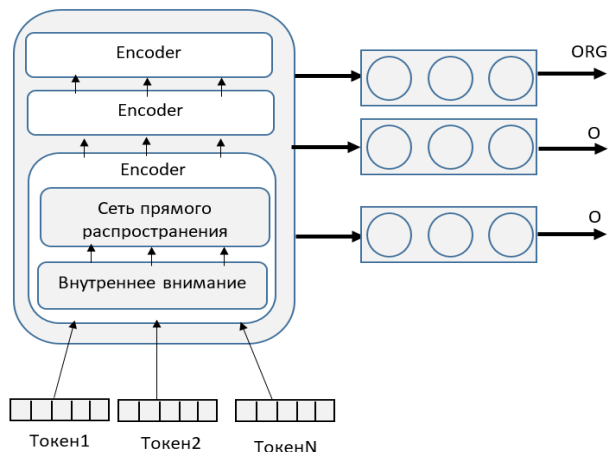


Рис. 1. Архитектура сети

2. Оценка эмоциональной окраски комментариев

При обработке данных новостных сайтов часто может быть полезно иметь представление об эмоциональной окраске комментариев к новости. С помощью нее можно оценить отношение людей к той или иной проблеме.

Для решения этой задачи будет применяться подход аналогичный с тем, что использовался для нахождения именованных сущностей.

В качестве предобученной модели выступает модель ELMo, представляющую из себя двухсвязную LSTM сеть [3] (рис. 2), обученную на корпусе русскоязычных сообщений в twitter. К ней добавляется сети прямого распространения, последний слой которой имеет функцию активации softmax. Полученная сеть обучается на новых размеченных данных, классифицируя сообщения на 3 класса: отрицательные, нейтральные, положительные.

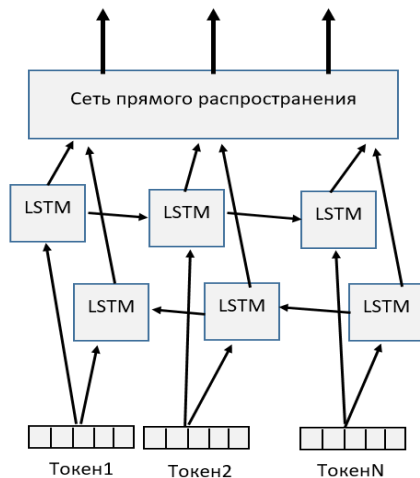


Рис. 2. Архитектура сети

3. Классификация текстов

Данная функциональность модуля производится для того, чтобы иметь возможность вести статистику по определенным категориям. Несмотря на то, что новостные сайты имеют в себе некую рубрикацию, необходимо классифицировать тексты по некоторому списку категорий из-за различия этих рубрик для разных сайтов, а также для удобства пользователя, так как он сможет сформировать список интересующих его тем.

В рамках разрабатываемого модуля для решения поставленной задачи был выбран подход с использованием мешка слов (bag-of-words) и меры значимости TF-IDF для кодирования слов и классификатора, использующего методы поверхностного машинного обучения.

Была сформирована обучающая выборка и для каждого текста из нее проводился этап предобработки текста, который включал в себя:

- удаление стоп-слов (часто употребительных слов, таких как союзы, предлоги и т.п.);
- лемматизация – приведение слова к его начальной форме;

Далее из слов очищенных текстов формируется словарь (мешок слов) и слова в тексте заменяются индексами в этом словаре. Затем выполняется частотный анализ с использованием метрики TF-IDF, после чего будет иметься разреженная матрица со значениями значимости каждого слова из “мешка” для каждой из категорий.

Следующим этапом идет обучение модели классификации.

Были реализованы 3 метода классификации представленных в таблице. В процессе разработки использовался язык программирования python и его библиотеки: numpy, pandas, Scikit-learn; среда разработки pyCharm. Данные, используемые в обучении, были получены с новостного сайта, размечены вручную и представляют собой dataframe, состоящий из текста новости и его категории.

Для каждого классификатора было произведено обучение, с использованием различных значений гиперпараметров, и тестирование на размеченных данных. Результаты работы каждого из них (для лучшего из наборов гиперпараметров) представлены в таблице. Была выбрана модель метода опорных векторов с использованием линейной функции ядра.

Таблица

Сравнение результатов классификации различных моделей

	precision	recall	F1-score
SVM с линейной функцией ядра	0.88	0.89	0.88
Логистическая регрессия	0.88	0.88	0.87
Случайный лес	0.83	0.84	0.83

Заключение

Был разработан рабочий программный модуль, для интеллектуального анализа текстовых данных новостных интернет ресурсов. Использование данного решения позволяет проводить обработку тестовой информации, получаемой из новостей для ее дальнейшего анализа и практического применения. В дальнейшем предполагается использование данного модуля в информационно-аналитической системе мониторинга и анализа новостных сайтов.

Список литературы

1. Ермаков, А. Е. Компьютерная лингвистика и интеллектуальные технологии / А. Е. Ермаков, С.А. Киселев. – М. : Наука, 2005. – 231 с.
2. Howard, J. Fine-tuned Language Models for Text Classification / J. Howard, S. Ruder // ArXiv:1801.06146. – 2018.
3. Schuster, M., Paliwal K. K. Bidirectional recurrent neural networks // IEEE Transactions on Signal Processing. – 1997. – v. 45. – №. 11. – P. 2673-2681.

Разработка персонализированного детектора ключевых точек лица

А. Д. Борисов

Студент бакалавриата

Д. И. Соломатин

Старший преподаватель

К. В. Климов

Консультант, Russian3DScanner

Введение

Современные стандарты индустрии кино и видеоигр требуют тщательности и точности в воспроизведении внешности настоящих людей (цифровых дублеров). В последние годы для создания высококачественных дублеров все чаще используются методы 4D сканирования. Для этого игра актера захватывается на видео с множества ракурсов. Для каждого захваченного кадра строится трехмерный скан, который точно воспроизводит форму лица актера. Необработанные сканы нельзя использовать для создания анимации, так как они имеют слишком большой размер и разную топологию. Поэтому полученные сканы приводят к одной топологии с использованием методов неригидной регистрации [1]. К сожалению, методы неригидной регистрации плохо справляются со сложными деформациями, особенно в области глаз и губ, и им требуется передать информацию о форме губ, глаз и других сложных регионов, извне. Данная работа посвящена разработке персонализированного алгоритма обнаружения ключевых областей лица, таких как губы и глаза, для задачи 4D регистрации.

1. Существующие решения

Для робастной работы неригидной регистрации необходимо иметь контуры глаз и губ. Эта проблема изначально решается использованием детектора на основе машинного обучения, в частности сверточной нейронной сети, которая по фронтальному изображению актера находит необходимые контуры, представленные набором точек. Но иногда сеть определяет точки контуров с большой погрешностью, что недопустимо для задачи неригидной регистрации.

Для решения этой проблемы можно обучать нейронную сеть под конкретного актера, но данная процедура слишком дорога. Если не брать в расчет подготовку большого количества примеров, которые нужно собрать и консистентно разметить, время обучения нейронной сети может занимать от нескольких часов, а в случае необходимости добавления дополнительных примеров в обучающую выборку данный процесс может сильно затянуться.

Поэтому необходимо найти решение, как исправлять результат, полученный сетью, обученной на множестве людей, под конкретного человека с минимальными трудозатратами. В решении данной проблемы могут помочь активные модели внешнего вида (ААМ) [2]. Оператор размечает ключевые кадры из последовательности изображений, которые впоследствии с помощью решения задачи оптимизации деформируют контур таким образом, чтобы область изображения, вырезанного по контуру, наилучшим образом совпадала с таким же изображением, но полученным в результате смешивания ключевых кадров, размеченных оператором.

Проблема этого подхода заключается в следующем. В рассматриваемой задаче важно иметь максимально возможное качество детектирования, поэтому исходные изображения имеют высокое разрешение, как минимум 96 dpi. А данная разработка [2] использует все изображение лица целиком, что очень сильно увеличивает время работы и ухудшает гибкость алгоритма. К тому же, оригинальная ААМ специализируется на детектировании любых лиц. В данной работе, предлагается ее модификация под персонализированный детектор.

2. Идея алгоритма персонализированного детектора

На рис. 1 представлен алгоритм работы персонализированного детектора. На вход алгоритм принимает набор ключевых кадров из последовательности кадров, представленных в виде пар (*Изображение, Точки контуров*), размеченных оператором. Эти кадры используются для обучения алгоритма, которое подразумевает создание двух математических моделей.

Изображения ключевых кадров обрабатываются особым способом для формирования математической модели текстурных деформаций. Эта модель позволит линейно смешивать изображения ключевых кадров с некоторыми весами. В то же время, точки с ключевых кадров используются для формирования математической модели пространства деформации точек, что позволит генерировать, регулируя ряд параметров, формы контуров, основываясь на ключевых кадрах. Этот процесс обучения алгоритма занимает считанные секунды, после чего он готов к работе.

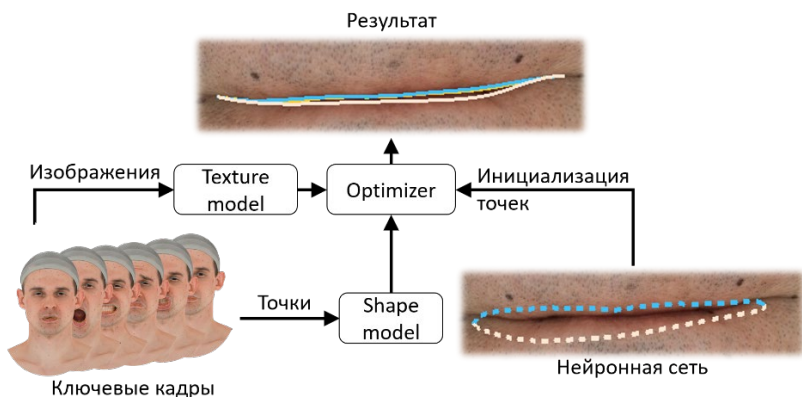


Рис. 1. Обзор подхода

На вход алгоритму, в момент работы, идет пара (*Изображение, Точки контуров*), где точки контуров получены с помощью нейронной сети. Результат нейронной сети используется в качестве начальной инициализации формы контура. Алгоритм сводится к решению определенной задачи оптимизации, после решения которой получается исправленный контур.

3. Математическая модель формы контура

Модель формы контура получается из контуров, взятых с ключевых кадров. Формы контуров предварительно ригидно выравниваются между собой. Затем вычисляются главные компоненты PCA и средняя форма контура. PCA, он же метод главных компонент, представляет собой ортогональное линейное преобразование, которое отображает данные из исходного пространство признаков в новое пространство меньшей размерности. При этом первая ось новой системы координат строится таким образом, чтобы дисперсия данных вдоль нее была бы максимальной из оставшихся осей. С помощью линейного смешивания между главными компонентами можно генерировать данные, похожие на те, которые использовались для вычисления главных компонент. В итоге, формула для вычисления формы контура выглядит следующим образом:

$$\varphi(\vec{w}) = \mathbf{C}\vec{w} + \vec{m}, \quad (1)$$

где \mathbf{C} – матрица из главных компонент, а \vec{m} – средняя форма контура. В качестве параметра используется \vec{w} – вектор весов смешивания главных компонент.

Для перемещения в пространстве к полученному контуру применяются глобальные трансформации, такие как глобальный поворот $\mathbf{R}(\alpha)$, где α – угол поворота и \mathbf{t} – смещение в пространстве. \mathbf{v} является вектором, однако для дальнейших вычислений представлен в виде матрицы, где каждой строке соответствуют координаты (x, y) . В результате чего получается итоговая формула математической модели для генерации произвольного контура.

$$\mathbf{v}(\mathbf{w}, \mathbf{t}, \alpha) = \mathbf{v}(\mathbf{w})\mathbf{R}^T(\alpha) + \mathbf{t}, \quad (2)$$

где регулируемые параметрами являются веса PCA, смещение и угол поворота в пространстве.

4. Извлечение патчей

Патч – это небольшое окно над изображением, хранящее набор координат \mathbf{p}_i вида (x, y) для последующего извлечения каждого пикселя. Важной идеей данного подхода является использование не всего изображения лица, а отдельных патчей изображений, которые лежат на сегментах контуров. Для их извлечения генерируется патч в начале координат. Для каждого сегмента контура вычисляется угол между ним и осью OX . Затем исходный патч поворачивается на этот угол и смещается на середину сегмента, которая вычисляется по следующей формуле.

$$\mathbf{T}_s(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 + \mathbf{v}_2}{2}, \quad (3)$$

где $\mathbf{v}_1, \mathbf{v}_2$ – это начальная и конечная точки конкретного сегмента. Таким образом, итоговая формула для получения значения из патча выглядит следующим образом:

$$\varphi(\mathbf{v}_1, \mathbf{v}_2, I)_i = I(\mathbf{R}_s(\mathbf{v}_1, \mathbf{v}_2)\mathbf{p}_i + \mathbf{T}_s(\mathbf{v}_1, \mathbf{v}_2)), \quad (4)$$

где I – специальная функция взятия значения пикселя из изображения, а \mathbf{R}_s – функция получения матрицы поворота для данного сегмента. Пример итогового результата расположения патчей и их значений отображен на рис. 2.

5. Математическая модель текстурной деформации

Для каждого ключевого кадра извлекаются значения патчей, расположенных на контурах. Значения каждого патча разворачиваются в вектор, таким образом получается матрица, где каждой строке соответствуют значения патчей для конкретного кадра.



Рис. 2. Пример расположения патчей (слева) и извлеченное изображение из-под патчей (справа)

В оригинальном подходе построения ААМ для создания модели текстурной деформации используют все изображение целиком, что уменьшает гибкость алгоритма и увеличивает время работы. Также используют РСА для смешивания между ключевыми кадрами. Однако в данном случае, когда имеются небольшие патчи, это приводит к получению изображений в виде равномерной серой массы, которые не подходят для работы. Исходя из этого, было принято решение в виде осуществления линейного смешивания патчей между ключевыми кадрами.

Таким образом, формула для генерации конкретного патча на основе ключевых кадров имеет следующий вид:

$$\Phi_T(i_1, i_2, \overset{\Gamma}{w}_T) = \sum_{j \in T} \overset{\Gamma}{w}_{T,j} [[\Phi(\overset{\Gamma}{v}_{T,i_1}, \overset{\Gamma}{v}_{T,i_2}, I_T) - \Phi_{i_1, i_2}] + \Phi_{i_1, i_2}], \quad (5)$$

где $\Phi_T(i_1, i_2, \overset{\Gamma}{w}_T)$ – функция линейного смешивания патчей из ключевых кадров для конкретного сегмента (i_1, i_2) , с весами смешивания $\overset{\Gamma}{w}$. Где i_1, i_2 – индексы точек сегментов, Φ_{i_1, i_2} – среднее значение конкретного патча для ключевых кадров, $\overset{\Gamma}{w}_T$ – веса смешивания патча между ключевыми кадрами.

6. Задача оптимизации

Необходимо задать функцию ошибки, которая будет показывать, насколько хорошо сгенерированный на целевом изображении контур похож на ключевые кадры.

На рис. 3 показан процесс вычисления ошибки. Из сгенерированного контура, который выдает математическая модель формы, извлекаются патчи с целевого изображения из-под каждого сегмента контура. В то же время, текстурная модель генерирует набор патчей. Далее патчи, полученные от текстурной модели, и патчи, полученные от сгенерированного контура, но с целевого изображения, сравниваются. Чем меньше разница между патчами с контура целевого

изображения и патчами, сгенерированными на основе ключевых кадров, тем лучше.

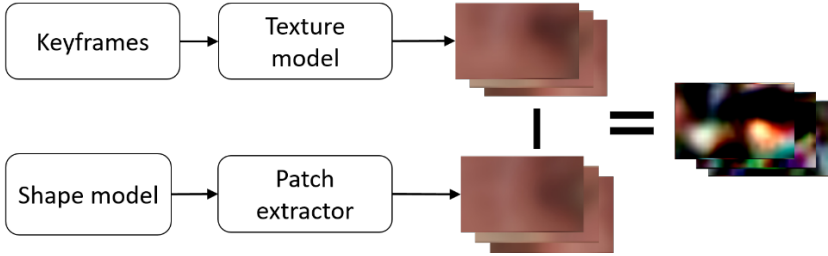


Рис. 3. Ошибка задачи

Итоговая формула функции ошибки выглядит таким образом:

$$E(\overset{r}{w}_s, \overset{r}{w}_T, \alpha, \overset{r}{t}) = \sum_{(i_1, i_2) \in \text{segs}} [\varphi(v_{\overset{r}{w}_s, \alpha, \overset{r}{t}, i_1}, v_{\overset{r}{w}_s, \alpha, \overset{r}{t}, i_2}, I) - \varphi_T(i_1, i_2, \overset{r}{w}_T)]^2. \quad (6)$$

где segs – множество сегментов конкретного контура. Переменными, которые можно регулировать, являются $\overset{r}{w}_s$ – веса смешивания компонент PCA для формы контура, α – угол поворота всего контура в пространстве, $\overset{r}{t}$ – смещение всего контура в пространстве, $\overset{r}{w}_T$ – веса смешивания патчей между ключевыми кадрами.

Таким образом, задача сводится к поиску переменных, при которых происходит минимизации следующего функционала.

$$\arg \min_{\overset{r}{w}_s, \overset{r}{w}_T, \alpha, \overset{r}{t}} E(\overset{r}{w}_s, \overset{r}{w}_T, \alpha, \overset{r}{t}). \quad (7)$$

Функция ошибки представлена суммой квадратов. Для решения этой задачи оптимизации используется метод Гаусс-Ньютона [3], где на каждой итерации считается смещение $\Delta \overset{r}{x}$ по значению параметров оптимизации. Таким образом новое значение параметром на каждой итерации вычисляется по следующей формуле:

$$\overset{r}{x}_{k+1} = \overset{r}{x}_k - \Delta \overset{r}{x} = \overset{r}{x}_k - \mathbf{H}^{-1} \mathbf{J}^T F(\overset{r}{x}), \quad (8)$$

где $F(\overset{r}{x})$ – невязки, а матрица Гессе аппроксимируется с помощью матрицы Якоби, а именно:

$$\mathbf{H} = \mathbf{J}^T \mathbf{J}. \quad (9)$$

Использование весов моделей текстурной деформации и точек контуров требует добавление определенных ограничений на их значения. Для этого в задачу оптимизации добавляется регуляризация по Тихонову, чтобы ограничить веса.

В результате решения данной задачи оптимизации получается набор параметров для модели формы контура, которые позволяют минимизировать ошибку и тем самым получить персонализированный контур на целевом изображении, основываясь на информации с ключевых кадров.

7. Итеративное улучшение результатов оптимизации

Нейронная может выдавать результаты, которые чрезвычайно далеки от требуемых в силу того, что обучающая выборка состояла из людей преимущественно молодой европеоидной внешности. Такая плохая инициализация сильно затрудняет работу персонализированного детектора.

Для улучшения робастности алгоритма используются Coarse-to-fine подход, идея которого состоит в постепенном масштабировании входных данных от меньшего масштаба к оригинальному, где в качестве инициализации каждого уровня выступает результат предыдущего уровня (на первом уровне – результат работы нейронной сети).

Входные данные масштабируются к низкому разрешению. Персонализированный детектор обучается и отрабатывает на этих данных для получения грубого определения контура. Затем результат идет на вход детектору следующего уровня, где входные данные масштабировались к более высокому разрешению. И так далее до достижения масштаба исходного изображения.

Такой подход позволяет получить более робастный результат в случае слишком плохой начальной инициализации нейронной сети. А также дает возможность добавить поддержку поддержки внутреннего контура губ при отсутствии такового у нейронной сети.

8. Результаты эксперимента

Итоговый результат работы изображен на рис. 4. Как можно увидеть, разработанный детектор позволяет получить более робастный результат по сравнению с нейронной сетью, при котором пользователю требуется производить разметку только ключевых кадров с экстремальными случаями. В случае необходимости пользователь может добавить проблемные кадры в ключевые и исправить результат.

Заключение

Таким образом, в результате работы удалось получить персонализированный детектор ключевых точек лица.

Данный алгоритм внедрен в программное обеспечение R3DS Track, поставляющиеся с программным комплексом для обработки 4D секвенций R3DS Wrap4D компании Russian3DScanner.

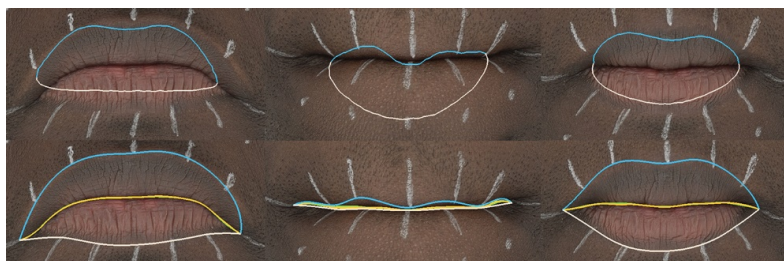


Рис. 4. Сравнение результатов нейронной сети (вверху) и персонализированного детектора (внизу)

Список литературы

1. Fernando, G. Mesh Wrap based on Affine-Invariant Coordinates / G. Fernando, M. Alonso // ACM SIGGRAPH computer graphics. – N.Y. : ACM, 2019. – P. 1-2.
2. Cootes, T. F. Active Appearance Models / T. F. Cootes, G. J. Edwards, C. J. Tatlor // ECCV. – N.Y., 1998 – P. 484-498.
3. Madsen, K. Methods For Non-Linear Least Squares Problems / K. Madsen, H. B. Nielsen, O. Tingleff. – 2-nd ed. – L. : IMM, 2004. – 30 p.

Решение задачи оптимального распределения ресурсов при планировании выпуска продукции в системе «1С:Предприятие 8.3»

С. Ю. Будаева

Студент бакалавриата

А. В. Копытин

Доцент

Введение

Большинство экономических и управленческих задач имеют, как правило, несколько вариантов решений. Среди них приходится отыскивать, по возможности, лучший (оптимальный) вариант, исходя из текущих условий. Для сравнения альтернативных решений выбирается некий критерий оптимальности, который записывается в математическом виде как функция нескольких переменных. Решить задачу оптимизации – значит найти такой набор значений переменных,

при котором функция достигает экстремума при соблюдении заданных ограничений. Если учесть, что количество переменных и ограничений может исчисляться сотнями или тысячами, то такая задача не кажется тривиальной.

Оперировать с большим объёмом данных стало возможным благодаря внедрению на предприятиях различных ERP-систем. Одним из лидеров российского рынка является система «1С:Предприятие». Ее объекты конфигурации – справочники, документы и регистры – структурировано хранят в себе большинство сведений, необходимых для построения математических моделей различных задач.

Данная статья посвящена рассмотрению решения задачи оптимального распределения ресурсов при планировании выпуска продукции и разработке для этих целей программного обеспечения, которое может быть интегрировано в систему «1С:Предприятие».

1. Задача оптимального распределения ресурсов при планировании выпуска продукции

Одной из часто решаемых задач линейного программирования является задача об оптимальном использовании ресурсов [1].

По условию задачи имеется m видов ресурсов в количестве b_1, b_2, \dots, b_m , которые могут быть использованы при производстве n видов изделий. Известны технологические коэффициенты a_{ij} ($i = 1, \dots, m; j = 1, \dots, n$), характеризующие количество i -го ресурса, необходимого для производства j -го изделия. Кроме этого, известна предполагаемая прибыль c_j от продажи единицы j -го изделия. Требуется составить такой план выпуска продукции, при котором прибыль от продажи всех изделий будет максимальной.

Обозначив через x_j количество j -го изделия, получим математическую модель, содержащую целевую функцию (1), ограничения (2) и граничные условия (3):

$$\max \sum_{j=1}^n c_j x_j, \quad (1)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = 1, \dots, m), \quad (2)$$

$$x_j \geq 0 \quad (j = 1, \dots, n). \quad (3)$$

Заметим, что в данной задаче на некоторые переменные может быть наложено требование целочисленности.

В классическом описании задачи предполагается, что все произведенные товары будут реализованы на рынке. Однако в реальности это происходит далеко не всегда, поэтому предлагается ввести дополнительное граничное условие:

$$x_j \leq d_j \quad (j = 1, K, n), \quad (4)$$

где d_j – максимальное количество j -го изделия, которое можно реализовать на рынке по прогнозу покупательского спроса.

Если рассматривать компанию, работающую по предварительному заказу, следует также ввести граничное условие:

$$e_j \leq x_j \quad (j = 1, K, n), \quad (5)$$

где e_j – количество j -го изделия, уже оплаченного покупателем.

Если $e_j \geq d_j$, то граничное условие (4) не применяется.

Таким образом получена задача поиска максимума линейной целевой функции с линейными ограничениями на множество допустимых решений. Подобные задачи решаются методами линейного программирования.

2. Хранение данных в базе «1С: Предприятие»

Все необходимые для решения задачи данные получают при помощи механизма запросов к соответствующим объектам конфигурации системы 1С (справочникам, документам, регистрам и др.). Наименования конкретных объектов, а также их наличие или отсутствие в разных конфигурациях могут различаться.

– Коэффициенты при переменных целевой функции (прибыль c_j от продажи единицы изделия) могут быть получены из данных регистров бухгалтерского учета.

– Виды ресурсов m , требуемые для производства изделий, могут храниться в виде реквизитов справочника «Номенклатура» или в соответствующих регистрах сведений. Там же могут храниться и количества того или иного ресурса, необходимого для производства единицы изделия (коэффициенты a_{ij} ограничений).

– Количество имеющихся в наличии ресурсов b_m на конкретный момент времени хранятся в соответствующем регистре накопления (подвид «остатки»).

– Максимальное количество произведенных изделий, которое можно реализовать на рынке, в базе системы в явном виде не присутствует и должно быть получено либо с помощью методов экспертных оценок, либо, что гораздо эффективнее, рассчитано

математическими методами экстраполяционного прогнозирования. Это единственный параметр математической модели, значение которого может быть скорректировано вручную лицами, принимающими решения (ЛПР). Полученные данные должны сохраняться либо в специально созданном для этого документе-регистраторе, либо напрямую в соответствующем регистре сведений.

3. Описание внешней обработки «ЗадачаРаспредРесурсов.epf»

Для выполнения различных действий над информацией в базе 1С (добавление, удаление, изменение данных и т.д.) применяется прикладной объект конфигурации – обработка 1С. Обработки могут встраиваться в конфигурацию (внутренние обработки) или храниться в виде отдельных файлов с расширением «*epf*» (внешние обработки). Для того, чтобы расширить функционал исходной конфигурации системы и при этом не менять ее структуру, авторами статьи была выбрана внешняя обработка в качестве инструмента работы с данными.

Внешняя обработка *ЗадачаРаспредРесурсов.epf* реализована в виде управляемого приложения 1С. Основные задачи, выполняемые обработкой, следующие:

- Получение при помощи механизма запросов к соответствующим объектам конфигурации системы 1С значений соответствующих величин.
- Расчет максимального количества изделий, которое можно реализовать на рынке (прогноз продаж).
- Предоставление ЛПР возможности для просмотра и ручной корректировки прогноза продаж.
- Решение полученной задачи линейного программирования симплекс-методом.
- Сохранение полученных результатов решения в регистре сведений ПланВыпускаИзделий.
- Создание отчета системы 1С – прикладного объекта конфигурации для обработки накопленной информации получения сводных данных в удобном для просмотра и анализа виде.

По своей сути любая обработка (как внутренняя, так и внешняя) является программой, выполняемой в системе «1С:Предприятие» и написанной на встроенном языке программирования 1С. Данный язык является предварительно компилируемым предметно-ориентированным языком высокого уровня [2]. В отличие от большинства современных языков программирования в нем отсутствует ряд возможностей, например, создание новых классов пользователя. Кроме того, на встроенном языке 1С сложно реализовать обработку большого объема данных с приемлемым быстродействием. Авторами статьи был

проведен сравнительный анализ времени выполнения задачи пузырьковой сортировки целочисленного массива размером 2000 элементов, реализованной на языке C# (среда разработки Visual Studio 2019 Community) и встроенном языке 1С (платформа «1С:Предприятие 8.3.14»). В первом случае время выполнения сортировки составило 13,9 миллисекунд, а во втором случае – 32,8 секунды.

Для таких прикладных задач, которые сложно или невозможно реализовать на встроенном языке 1С, наиболее универсальным средством является «Технология создания внешних компонент»[3]. Данная технология позволяет создавать сторонние динамически подключаемые библиотеки для тесного взаимодействия с системой «1С:Предприятие», расширяя ее функциональность. Один из способов разработки внешних компонент – использование технологии COM (Component Object Model). Это метод разработки двоичных исполняемых файлов, которые предоставляют приложениям и другим компонентам необходимые сервисы в архитектуре «клиент-сервер».

Для решения оптимизационной задачи были разработаны две внешних компоненты Predictor.dll и Solver.dll с использованием среды разработки Visual Studio 2019 Community и языка программирования C#.

4. Описание внешней компоненты «Predictor.dll»

Для количественного исследования рынка сбыта анализируются данные о продажах за прошедший период времени на основании документа «Продажи» или аналогичного. По своей сути эти данные являются дискретными временными рядами. Для прогнозирования используется адаптивная модель, способная менять свои параметры и структуру, приспосабливаясь к изменению данных ряда (выбросам, скачкам, смене тренда). Основой адаптивной модели являются три предиктора – полиномиальные модели (нулевого, первого и второго порядков), основанные на методе экспоненциального сглаживания с изменяемым параметром адаптации по методу У. Чоу [4]. Итоговый прогноз вычисляется как взвешенная сумма прогнозов трех предикторов. Веса прогнозов обратно пропорциональны величинам ошибок. Пример обращения к внешней компоненте приведен в листинге 1.

Листинг 1

Пример обращения к внешней компоненте Predictor.dll

```
Процедура Прогнозировать (Команда)
ИсходныйРяд = "1 2 3 4 5 6 17 18 19 20 21 22 25";
МойОбъект = Новый СОМОбъект("Predictor.MyClass");
Прогноз = МойОбъект.Predict(ИсходныйРяд);
. . .      КонецПроцедуры
```

5. Описание внешней компоненты «Solver.dll»

Внешняя компонента Solver.dll предназначена для решения задач линейного программирования симплекс-методом, который основан на последовательном переходе от одного допустимого решения (опорного плана) задачи к другому, при котором значение целевой функции возрастает. Такой переход возможен, если известен некоторый исходный опорный план.

Когда все ограничения представлены неравенствами вида «меньше или равно», таким исходным опорным планом является нулевой вектор. При ограничениях вида «равно» и «больше или равно» следует применить иной способ. Во внешней компоненте Solver.dll реализован двухэтапный метод решения с использованием симплекс-таблиц [5] в соответствии с алгоритмом:

1. Построение искусственного базиса.
2. Поиск начального недопустимого решения.
3. Переход к начальному допустимому решению.
4. Переход к оптимальному решению.

Работу с внешней компонентой рассмотрим на простом примере из [5]. Краткое условие задачи: «Цех выпускает изделия: корпуса и задвижки. На один корпус расходуется 20 кг алюминия, 10 кг стали и 5 кг пластмассы, а на задвижку – 5 кг алюминия, 5 кг стали и 20 кг пластмассы. За смену цех может израсходовать не более 200 кг алюминия, 250 кг стали и 500 кг пластмассы. Выпуск одного корпуса приносит прибыль 100 единиц, одной задвижки – 300 единиц. Требуется составить оптимальный план работы цеха.»

Пример обращения к внешней компоненте для решения этой задачи приведен в листинге 2.

Листинг 2

Пример обращения к внешней компоненте Solver.dll

```
Процедура Решить (Команда)
    ЦФ = "100 300";           // Целевая функция
    Огр = "20 10 5 5 5 20";  // Ограничения (в строку)
    Зн = "1 1 1";           // Знаки: 1 <=, e =, g >=
    Лим = "200 250 500";    // Лимит склада
    Экстр = "max";          // Вид экстремума max/min
    МойОбъект = Новый СОМОбъект("Solver.MyClass");
    Результат = МойОбъект.Сolve (ЦФ, Огр, Зн, Лим, Экстр);
    . . . КонецПроцедуры
```

В результате работы внешней компоненты в регистре сведений *ПланВыпускаИзделий* сохраняются: значение предполагаемой прибыли, количество планируемых к выпуску изделий, статус ресурсов

(дефицитные/недефицитные) и теневые (скрытые) цены ресурсов. На основании данных регистра создается отчет в требуемой форме.

Заключение

В статье было рассмотрено решение задачи оптимального распределения ресурсов при планировании выпуска продукции средствами платформы «1С:Предприятие 8» с различными конфигурациями. Аналогичный подход может применяться для решения других оптимизационных задач: о назначениях, об использовании мощностей оборудования, о раскрое материалов, о составлении рациона питания, транспортной задачи и других.

При этом следует отметить, что созданные для этих целей внешние компоненты могут использоваться не только совместно с платформой 1С, но и с иными программными приложениями при решении задач линейного программирования симплекс-методом, а также при прогнозировании временных рядов.

Список литературы

1. Экономико-математические методы и прикладные модели : учебное пособие для вузов / В. В. Федосеев [и др.] ; под. ред. В. В. Федосеева. – М. : ЮНИТИ, 1999. – 391 с.
2. Радченко, М. Г. 1С: Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – М. : ООО «1С-Паблишинг», 2013. – 965 с.
3. Технология создания внешних компонент [Электронный ресурс] : Информационно-технологическое сопровождение (1С: ИТС). – Режим доступа : <https://its.1c.ru/db/metod8dev#content:3221:hdoc>
4. Лукашин, Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов : учебное пособие / Ю. П. Лукашин. – М. : Финансы и статистика, 2003. – 416 с.
5. Смородинский, С. С. Оптимизация решений на основе методов и моделей математического программирования : учебник для вузов / С. С. Смородинский, Н.В. Батин. – Мн. : БГУИР, 2004. – 237 с.

Разработка прототипа приложения для автоматизации проверки решений задач по программированию в учебном процессе

М. С. Верещагина

Студент

Д. И. Соломатин

Старший преподаватель

Введение

В современном мире проверка успеваемости учащихся является неотъемлемой частью процесса обучения. Но в эпоху развития информационного общества мы все чаще наблюдаем, как в своей деятельности человечество для повышения эффективности использует различные автоматизирующие системы, которые могут заменить традиционные методы контроля и преподавания.

На сегодняшний день в свободном доступе находится несколько систем онлайн-тестирования решений задач по программированию, однако все они ориентированы на тестирование олимпиадных заданий: олимпиады проводятся не часто, задачи там сложные и их не много. В учебном процессе главный упор делается не на сложность задач, а на их количество и быстроту подготовки. Задачи могут быть не сложными, тестов к ним может быть не много, однако в существующих тестовых системах преподавателю придется тратить значительное количество времени на подготовку даже самых простых заданий, что затрудняет использование таких систем в учебном процессе.

Поэтому в данной работе рассматривается задача разработки прототипа приложения для автоматизации проверки решений задач по программированию, позволяющего упростить процесс подготовки задач для преподавателя по сравнению с уже существующими системами с целью дальнейшей интеграции в учебный процесс.

1. Требования к разрабатываемому приложению

Целью данной работы является разработка прототипа веб-приложения для автоматизации решений задач по программированию, обладающего следующими функциональными возможностями:

- авторизация пользователей в системе с разделением ролей на преподавателя и учащегося;

– предоставление учащемуся условия задачи и возможности отправления своего собственного решения в виде кода на одном из языков программирования с незамедлительным получением результатов автоматической проверки;

– помимо стандартных добавления и удаления задач предоставить преподавателю возможность их редактирования через приложение без использования сторонних ресурсов.

Одними из главных требований к разрабатываемому приложению являются расширяемость функциональности в виде добавления новых языков программирования в уже реализованный прототип без необходимости полного переделывания системы, а также простота подготовки преподавателем задач с тестами.

2. Оформление файла с условием задачи

Для простоты и удобства подготовки задач преподавателем для одной задачи будет подготавливаться только один файл, содержащий условие задачи в формате Markdown [1] и входные тесты (листинг 1).

Листинг 1

Пример оформления файла с условием задачи, решение которой представляет собой консольную программу

```
# Задача 1. Вложенные циклы. Простые алгоритмы #
На вход программы будет передаваться только s - размер фигуры
(s >= 1).
Необходимо напечатать в консоле следующую фигуру (в примере
ниже: s = 6):
*****
 *$$$*
  $$$*
   $*$
    **
     *
:checker
:input
6
:output
*****
 *$$$*
  $$$*
   $*$
    **
     *
```

С двоеточия начинаются все разделы. Разделитель `:checker` отделяет условие задачи, которое будет отображаться учащемуся, от входных и выходных (разделители `:input` и `:output`) данных, используемых при тестировании и которые студент не видит.

Если преподаватель хочет, чтобы решение учащегося представляло собой не полную программу, а лишь реализацию функции, которая затем будет вызываться в программе (листинг 2), написанной преподавателем, то в таком случае используется разделитель `:program_с` указанием расширения файла для компиляции, соответствующего языку реализации, который отделяет код программы для тестирования от остального содержания файла задачи. В то место в коде, где должна располагаться реализация функции учащегося следует вставить `$student_code`.

Листинг 2

Пример оформления файла с условием задачи, решение которой представляет собой функцию

```
# Задача 2. Условный оператор. Функции #
Проверить, можно ли вписать в треугольник со сторонами A, B, C
круг с радиусом R (заодно проверить возможность существования
треугольника).
Решение задачи реализовать в виде отдельной функции Solution,
печатающей ответ в консоль, и предоставить ее в качестве
решения:
void Solution(double a, double b, double c, double rad)
:program_cpp
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream>
using namespace std;
$student_code
int main() {
    double a,b,c;
    double rad;
    cin >> a;
    cin >> b;
    cin >> c;
    cin >> rad;
    Solution(a, b, c, rad);
    return 0;
}
:checker
:input
2 3 5 1
:output
Не существует
:input
3 4 5 1
:output
Можно
```

В некоторых задачах преподавателю вместо правильного ответа гораздо проще указать правильное решение (листинг 3). В таком случае, он может с помощью разделителя: `:correct_solution_` с указанием расширения файла для компиляции, соответствующего языку реализации, оставить верное решение задачи, которое будет использоваться при проверке решения учащегося.

Листинг 3

*Пример оформления файла с условием задачи, где указано
правильное решение*

```
# Задача 3. Оператор присваивания #
Вычислить площадь окружности, вписанной в прямоугольный
равнобедренный треугольник, если известна длина его
гипотенузы.
:correct_solution_cpp
#define _USE_MATH_DEFINES
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream>
using namespace std;
int main() {
    double a, s;
    cin >> a;
    s = M_PI*M_PI*a*(1-sqrt(2)/2);
    cout << s;
    return 0;
}
:checker
:input
5
:input
7
:input
2
```

Можно задать несколько шаблонов программы на разных языках, что делает данный файл с условием задачи универсальным для различных языков программирования.

3. Реализация

Архитектура системы представлена на рис. 1. Система реализована в виде веб-сервера на основе клиент-серверной архитектуры. Вся клиентская часть приложения была реализована с помощью HTML и CSS. Серверная сторона приложения написана на языке Python, в ее основе лежит микрофреймворк Flask [2]. Сервер предназначен для установки в среде ОС Linux и для использования веб-сервером Apache или Nginx.

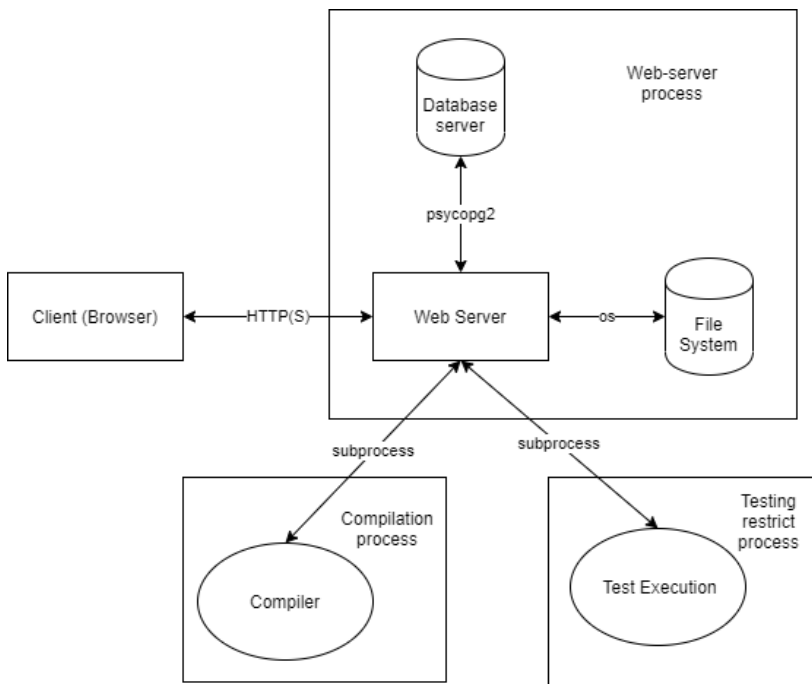


Рис. 1. Архитектура приложения

Все данные системы, за исключением файлов с задачами, хранятся в базе данных на сервере приложения. В качестве реляционной СУБД была выбрана PostgreSQL [3]. За связь с базой данных отвечает расширение SQLAlchemy. Файлы с условиями задач хранятся в файловой системе, для взаимодействия с которой веб-сервер использует модуль os.

Механизм тестирования присылаемого кода был реализован в виде bash-скриптов, которые привязываются к языку программирования, что позволит в дальнейшем расширить систему на неограниченное количество языков. Ввод и вывод данных скриптов перенаправлен в соответствующие файлы.

Тестирование состоит из двух частей: компиляции и запуска. Компиляция заключается в запуске процесса компиляции текста программы с помощью модуля subprocess. В качестве аргумента командной строки веб-сервер передает исходный файл с отправленным кодом учащегося. Для тестирования создается отдельный процесс с ограниченными правами для вызова bash-скрипта с запуском в этом

процессе скомпилированного кода. Такой подход позволяет накладывать ограничения на исполняемые код, связанные с требованиями безопасности.

4. Интерфейс пользователя

Интерфейс системы проверки решений задач по программированию должен быть максимально простым и понятным, чтобы не отвлекать преподавателей и учащихся от самих заданий.

Для ускорения процесса прототипирования был использован фреймворк Bootstrap [4].

После успешной авторизации в системе учащемуся будет отображена страница, содержащая как условие задачи, выбранной для него преподавателем, так и текстовое поле для ввода решения задачи (текста программного кода).

Внешний вид этой страницы представлен на рис. 2.

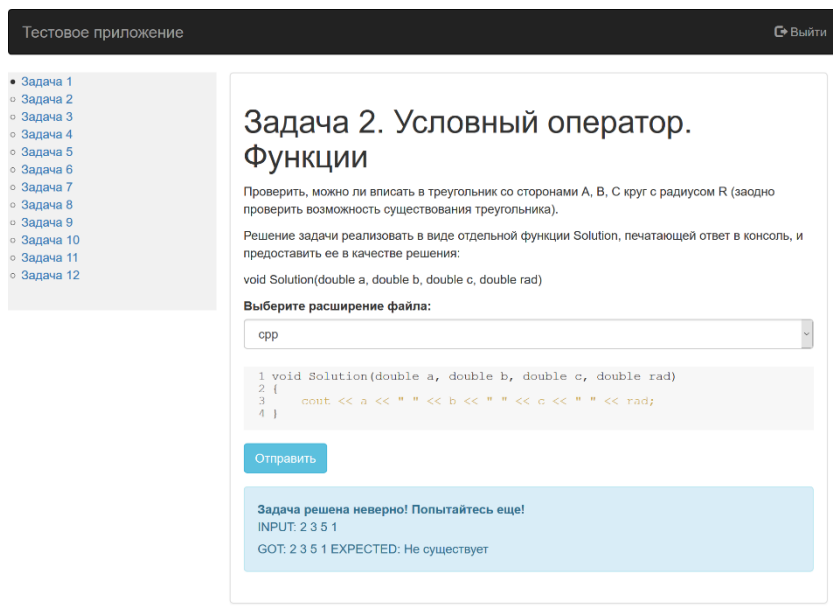


Рис. 2. Интерфейс учащегося

Над текстовым полем для ввода решения задачи находится выпадающий список с доступными расширениями файлов для компиляции.

Внизу страницы отображаются результаты последнего тестирования: верно ли была и решена задача, и если нет, то на каком этапе произошла ошибка: компиляции или на одном из тестов.

Если основной целью разработки интерфейса учащегося является оформление процесса тестирования решений, то основной целью разработки интерфейса преподавателя является предоставление удобного инструмента для редактирования задач для студентов. Интерфейс преподавателя представлен на рис. 3.

Учащийся не имеет доступа к интерфейсу преподавателя.

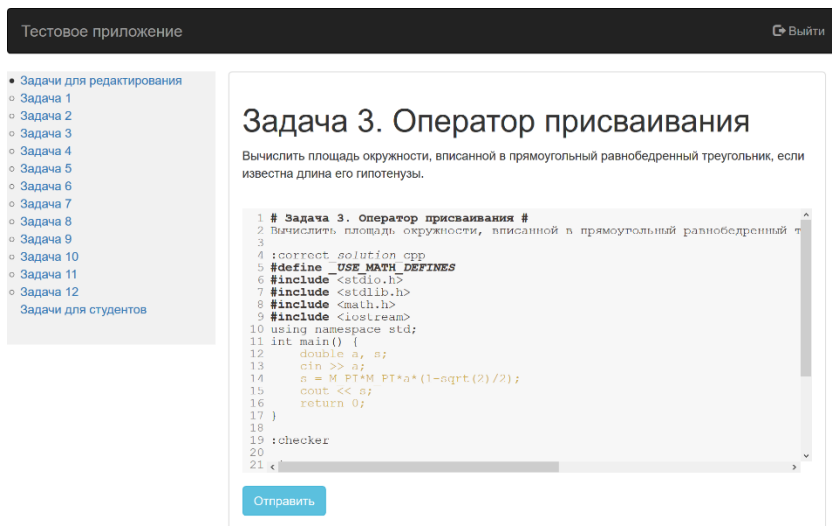


Рис. 3. Интерфейс преподавателя

Заключение

В ходе работы был разработан прототип веб-приложения, позволяющего экономить время преподавателя для подготовки массовых тестовых заданий и который позволяет широко применять автоматическое тестирование в образовательном процессе.

Для интеграции в систему были выбраны и описаны реальные задачи, предлагаемые студентам первого курса. На данный момент были протестированы консольные программы на языках C/C++.

В ближайших планах улучшение дизайна приложения, оптимизация процедуры подготовки простых задач для сведения к минимуму времени на подготовку тестирования, повышение контроля уровней доступа для расширения возможности применения системы,

добавление функционала контроля результатов тестирования преподавателем.

После доработки планируется постепенное внедрение системы в учебный процесс на первом курсе ФКН в ВГУ.

Список литературы

1. GitHub Repository Python-Markdown [Электронный ресурс]. – Режим доступа : <https://github.com/Python-Markdown/markdown>
2. Grinberg, M. Flask Web Development / M. Grinberg. – First Edition. – Cambridge : O'Reilly, 2014. – 238 p.
3. PostgreSQL documentation [Электронный ресурс]: база данных. – Режим доступа : <https://www.postgresql.org/docs/>
4. Bootstrap документация [Электронный ресурс]. – Режим доступа : <https://bootstrap-4.ru>

Приближенное построение двойственных систем с помощью рядов Фурье

В. П. Волгина

Студент бакалавриата

Е. А. Киселев

Доцент

Введение

Разложение в ряд по некоторому набору функций является важным инструментом в различных областях науки и техники. На этом основаны многие алгоритмы цифровой обработки сигналов, математический аппарат квантовой теории, приближенные методы решения дифференциальных уравнений, и т. д.

Несомненным преимуществом ортогональных систем является наличие универсального метода разложения по ним. Тем не менее, семейства функций, возникающие во многих математических моделях, зачастую являются неортогональными. В данной работе предлагается новый приближенный способ построения двойственных систем, которые являются важным инструментом для разработки алгоритмов разложения по неортогональным семействам. В качестве примера он будет реализован для систем целочисленных сдвигов функций Гаусса и

Лоренца, играющих, в частности, важную роль в атомной и ядерной физике [1, 2].

1. Постановка задачи и основные соотношения

В данной работе рассматривается пространство $L_2(R)$, скалярное произведение в котором обычно задается следующим образом:

$$(f, g) = \int_{-\infty}^{\infty} f(x) \overline{g(x)} dx, \quad (1)$$

где $\overline{g(x)}$ – комплексное сопряжение функции $g(x)$. Пусть дано семейство целочисленных сдвигов $\phi(\cdot - k)$, $k \in Z$, некоторой функции $\phi \in L_2(R)$. Сформулируем определение двойственной системы для этого частного случая.

Определение [3, 4]. Два набора функций $\phi(\cdot - k)$, $k \in Z$, и $\psi(\cdot - k)$, $k \in Z$, образуют двойственную систему в пространстве $L_2(R)$, если выполняется соотношение

$$(\phi(\cdot - m), \psi(\cdot - n)) = \int_{-\infty}^{\infty} \phi(x - m) \overline{\psi(x - n)} dx = \delta_{mn}, m, n \in Z, \quad (2)$$

где δ_{mn} – символ Кронекера.

Отметим, что часто используется также термин «биортогональная система». В частном случае, когда функции $\phi(\cdot - k)$, $k \in Z$, попарно ортогональны, наборы $\phi(\cdot - k)$, $k \in Z$, и $\psi(\cdot - k)$, $k \in Z$, совпадают.

Соотношение (2) можно записать более компактно. Для этого выполним замену переменной $t = x - n$ и обозначим $k = m - n$:

$$\int_{-\infty}^{\infty} \phi(t - k) \overline{\psi(t)} dx = \delta_{0k}, k \in Z. \quad (3)$$

Для построения двойственной системы требуется найти функцию $\psi(t)$, удовлетворяющую равенству (3).

Будем искать $\psi(t)$ в виде конечного ряда Фурье

$$\psi(t) = \sum_{m=-M}^M d_m e^{imht}, \quad (4)$$

где M – натуральное число, $h = \frac{2\pi}{2M+1}$, d_m – неопределенные коэффициенты. Подставив формулу (4) в условие (3), получим следующее соотношение:

$$\sum_{m=-M}^M \overline{d_m} \int_{-\infty}^{\infty} \phi(t-k) e^{-imht} dt = \delta_{0k}. \quad (5)$$

Выполнив далее замену переменной $x = t - k$, придем к равенству

$$\sum_{m=-M}^M \overline{d_m} e^{-ikmh} \int_{-\infty}^{\infty} \phi(x) e^{-imhx} dx = \delta_{0k}. \quad (6)$$

Если задать преобразование Фурье в виде

$$\hat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix\xi} dx, \quad (7)$$

то соотношение (6) можно кратко записать следующим образом:

$$\sqrt{2\pi} \sum_{m=-M}^M \overline{d_m} e^{-ikmh} \hat{\phi}(mh) = \delta_{0k}. \quad (8)$$

Обозначим $c_m = \sqrt{2\pi} \overline{d_m} \hat{\phi}(mh)$. Тогда формула (8) примет вид

$$\sum_{m=-M}^M c_m e^{-ikmh} = \delta_{0k}. \quad (9)$$

Если число M конечно, то равенство (9) не может быть выполнено при всех целых значениях k . В этом легко убедиться, если применить к формуле (9) дискретное преобразование Фурье [5]. Тем не менее, если положить $c_m = \frac{1}{2M+1}$, то соотношение будет справедливо при целых значениях $k \in [-2M, 2M]$. Поскольку на практике сигналы всегда имеют конечную длину, этого вполне хватит для проведения расчетов, если выбрать параметр M достаточно большим.

В итоге мы получаем следующий результат: если задать функцию $\psi(t)$ формулой

$$\psi(t) = \frac{h}{(2\pi)^{3/2}} \sum_{m=-M}^M \frac{e^{imht}}{\hat{\phi}(mh)}, \quad (10)$$

где $h = \frac{2\pi}{2M+1}$, то будет справедливо равенство

$$\int_{-\infty}^{\infty} \phi(t-k) \overline{\psi(t)} dx = \delta_{0k}, k = -2M, -2M+1, \dots, 2M. \quad (11)$$

2. Двойственные системы для семейств целочисленных сдвигов функций Гаусса и Лоренца

Далее мы рассмотрим процедуру построения двойственных систем по формуле (10) на примере систем целочисленных сдвигов функций Гаусса и Лоренца.

Пусть $\phi(t)$ – функция Гаусса:

$$\phi(t) = \exp\left(-\frac{t^2}{2\sigma^2}\right), \sigma > 0. \quad (12)$$

Ее образ Фурье имеет вид

$$\hat{\phi}(\xi) = \sigma \exp\left(-\frac{\sigma^2 \xi^2}{2}\right). \quad (13)$$

Для построения двойственной системы формулу (13) необходимо подставить в равенство (10). К сожалению, найти аналитически сумму в формуле (10) в случае функции Гаусса не удастся. На рис. 1 в качестве примера показан график функции $\psi(t)$ при $\sigma = 0.5$ и $M = 10$.

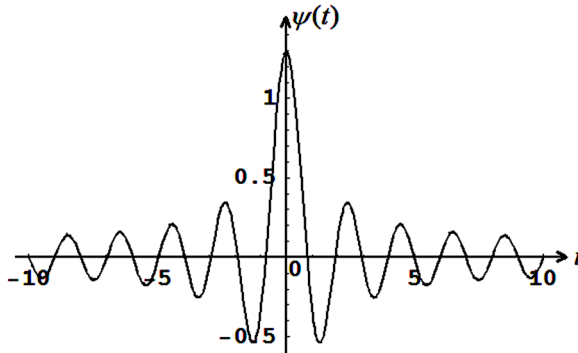


Рис. 1. График $\psi(t)$ в случае функции Гаусса при $\sigma = 0.5$ и $M = 10$

Пусть $\phi(t)$ – функция Лоренца:

$$\phi(t) = \frac{\sigma^2}{\sigma^2 + t^2}, \sigma > 0. \quad (14)$$

Ее образ Фурье определяется выражением

$$\hat{\phi}(\xi) = \sigma \sqrt{\frac{\pi}{2}} e^{-\sigma|\xi|}. \quad (15)$$

Если подставить соотношение (15) в формулу (10), то после несложных, но достаточно громоздких преобразований получится следующий результат:

$$\psi(t) = \frac{h}{2\pi^2\sigma} \left(\frac{1 - a^2 + 2a^{M+2} \cos(Mht) - 2a^{M+1} \cos((M+1)ht)}{1 + a^2 - 2a \cos(ht)} \right), \quad (16)$$

где $a = e^{h\sigma}$. Таким образом, в случае функции Лоренца удастся аналитически получить явную формулу для $\psi(t)$. На рис. 2 в качестве примера показан график функции $\psi(t)$ при $\sigma = 0.5$ и $M = 10$.

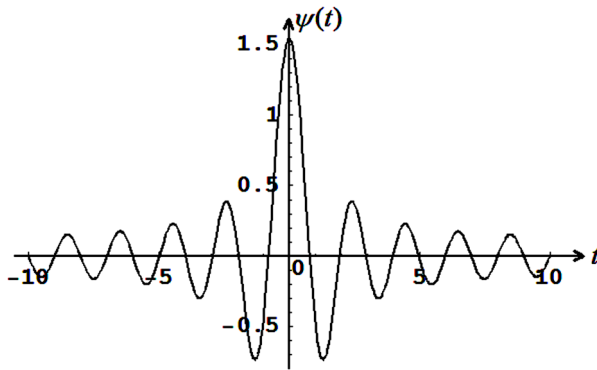


Рис. 2. График $\psi(t)$ в случае функции Лоренца при $\sigma = 0.5$ и $M = 10$

Заключение

В данной работе предложен новый приближенный способ построения двойственных систем для семейств целочисленных сдвигов. Его достоинством является отсутствие в формулах бесконечных рядов, в отличие, например, от метода, описанного в монографиях [3] и [4]. В ряде случаев разработанный нами способ позволяет получить аналитически достаточно простые соотношения.

В дальнейшем планируется рассмотреть и другие практически важные системы целочисленных сдвигов, использовать разработанный математический аппарат для создания новых алгоритмов цифровой обработки сигналов и провести вычислительные эксперименты с реальными сигналами.

Список литературы

1. Дробышев, А. И. Основы атомного спектрального анализа / А. И. Дробышев. – Санкт-Петербург : Изд-во Санкт-Петербургского университета, 1997. – 198 с.
2. Мухин, К. Н. Экспериментальная ядерная физика: в 3 т. Т. 2 Физика ядерных реакций. / К. Н. Мухин. – Санкт-Петербург : Лань, 2009. – 336 с.
3. Новиков, И. Я. Теория всплесков / И. Я. Новиков, В. Ю. Протасов, М. А. Скопина. – Москва : ФИЗМАТЛИТ, 2005. – 616 с.
4. Чуи, Ч. Введение в вейвлеты / Ч. Чуи ; пер. с англ. Я. М. Жилейкина. – М. : Мир, 2001. – 412 с.
5. Сергиенко, А. Б. Цифровая обработка сигналов / А. Б. Сергиенко. – Санкт-Петербург : Питер, 2006. – 751 с.

Исследование игрового поведения участников on-line игр на основе онтологического подхода

Т. С. Гаршин

Студент бакалавриата

А. Ю. Иванков

Ассистент

Введение

Главными задачами Game Analytics является отслеживание интересов игроков в ММО (Massively Multiplayer Online Game). Для этого существует много различных подходов к классификации типов поведения игроков с целью сегментации их для понимания тенденций динамики игры и ее планирования разработчиками [1, 2].

Для построения модели семантического описания и исследования поведения игроков предлагается использовать онтологический подход. Это направление развивалось в ряде работ исследователей в Game Analytics [3-5], результаты которых были применены в разрабатываемом проекте.

Архитектура системы Game Analytics

Для реализации системы Game Analytics, поддерживающей возможность семантического (смыслового) понимания и интерпретации

процесса принятия решений участниками онлайн игры, предлагается следующая архитектура, представленная на рис. 1.

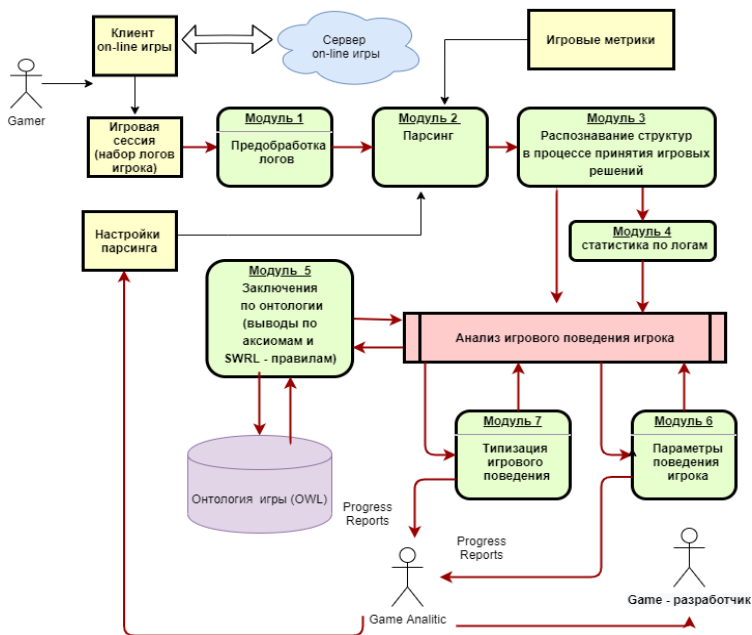


Рис. 1. Архитектура системы семантической Game Analytics

Исходными данными для анализа являются файлы логов игровых сессий участников MMO, которые ведутся на платформе on-line игры (или клиентском приложении). Рассмотрим модули и процессы в архитектуре предлагаемой системе анализа игрового поведения – Game Analytics.

Игровая сессия. Игровая сессия протоколируется в игровых логах, которые содержат структурированные данные о событиях, происходящих в игре: получаемых заданиях; информации, поступающей из разных источников; действиях игрока, его выборах и решениях; действиях по отношению к нему игровых персонажей: других участников MMO (людей играющих за персонажей) взаимодействующих с ним.

В логах также фиксируется текущая статистика по игровым достижениям (уровень, опыт, решение игровых задач, добытые знания и предметы, чаты с другими персонажами и др.). Размер файлов логов

зависит от протяженности во времени игровой сессии. В некоторых онлайн играх текущие игровые процессы фиксируются параллельно в нескольких лог-файлах: боевые, взаимодействия и др. Формат лог-файла привязан к конкретной игре, кроме того он содержит фрагменты текста на естественном языке, который поддерживается игрой. Тексты могут касаться фиксации игровых действий, диалоги с персонажами и людьми-игроками.

Предобработка и парсинг логов. Лог файлы одного игрока могут быть представлены набором файлов, которые представляют собой основу для анализа и смысловой интерпретации игрового поведения. Данные могут одновременно записываться в несколько логов, содержащих отдельные ракурсы игры (бои, переговоры с участниками, общие и др.) В связи со сложностью организации их структуры, они должны быть предобработаны парсером с целью извлечения и структурирования данных, хранящихся в логах (с учетом последовательности игровых сессий). Алгоритм разбора логов парсера ориентирован на структуру логов анализируемой игры и учитывает ракурс анализа. Выделенные структурированные данные могут быть дополнены статистикой, собранной по игровому логу.

В логах также фиксируется набор средств, с помощью которых игрок взаимодействует и предпринимает действия в игре. Они представлены набором возможностей для проявления реакций (действий) игроком, предусмотренных в игре (например, ввод сообщений, осуществление выборов и манипулирования игровыми сущностями и др.).

Для каждого способа действий и реакции можно выделить определенные маркеры (атрибуты), которые могут использоваться для мониторинга событий, отраженных в логах, и их разбора парсером. Например, могут быть выделены маркеры: постановка задания – результат выполнения задания; выбрано действие – получен результат действия и др.

Модуль распознавания структуры процесса принятия игровых решений. Игровой процесс представляет собой цикл деятельности: последовательности ситуаций, в которых игроки стараются достичь цели (решить задачи) через выбор средств достижения целей (им доступных или известных действий), и затем, получают в явном виде результаты своих действий на уровне тактики и стратегии в игре (рис.2). Извлекаемые из логов цепочки: Цель-Действие-Результат (ЦДР) – могут иметь иерархические, сложные структуры (рис. 3).

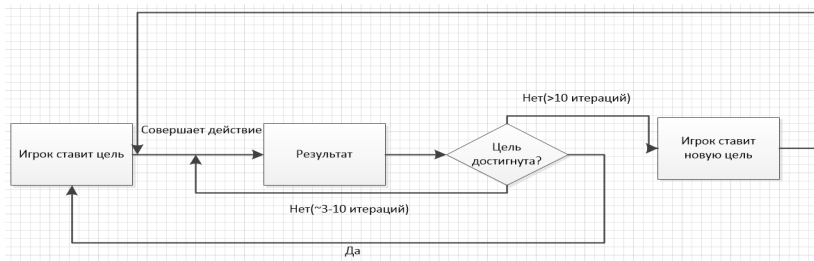


Рис. 2. Пример цикла принятия решений игроком

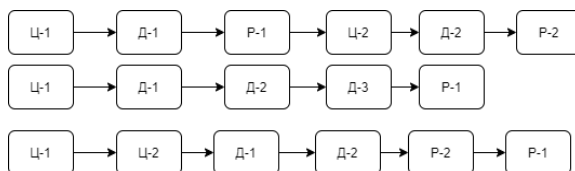


Рис. 3. Пример структур процесса принятия игровых решений

Цели могут быть явно сообщены игроку на шаге игры или могут быть не определены для игрока (игрок сам выбирает цель – построить дом, посадить репку, пойти воевать, поговорить и др). Они могут иметь разный уровень приоритета, степень детализации в игровой сессии. Целью наивысшего уровня может быть победа в игре или играть как можно лучше. Цели низкого уровня могут быть эквивалентны действиям. Для них можно определить взаимно-однозначное соответствие между целями и манипулированием сущностями через действия.

Время достижения цели может быть коротким (пересечь реку) и цепочка ЦДР будет короткой. Цель может быть долгосрочной (разгадать тайну) и цепочка ЦДР будет длинной, возможно, в ее состав могут входить подструктуры, соответствующие решениям промежуточных задач игроком. Понимание целей игрока и степени их достижения, оценка эффективности выбираемых действий обеспечивает обратную связь в игре. В связи с этим, необходимо распознавание структур ЦДР в процессе принятия игровых решений. Для распознавания требуются индикаторы – маркеры, позволяющие идентифицировать элементы структур ЦДР (например, для оценки достижения цели: *результат=цели, уровень успеха=цели* и т. д.).

Кроме того для игр характерны формы сегментации игрового процесса: «Spatial» (пространственные сегменты, например фрагменты карты игрового мира), «Temporal» (временная сегментация),

«Challenge» (игровые задания, миссии). Для каждой из этих форм сегментации можно выделить более мелкие (короткие) элементы, позволяющие отслеживать игровой процесс – «Checkpoint» (контрольные точки), соответствующие позициям проверки достижения результата.

Игровые сессии, принадлежащих одному игроку, могут иметь переменную длину, а выделение в них структур игровых логов, последующий их анализ и сравнение (в том числе историческое) будет одной из характеристик индивидуальности поведения игрока.

Статистика достижений игрока может быть собрана на основе данных об игровой сессии, представленной в логах, так и оценена внешне. Интересной является статистика по каждой игровой сессии отображающая количество: полученного опыта, взятых\выполненных заданий, убитых мобов, активность в чате, занятия ремеслом, получение денег.

Модуль построения заключений по онтологии. Для построения модели семантического описания и исследования поведения игроков предлагается использовать онтологический подход. Для его реализации в структуру архитектуры системы семантической Game Analytics включена онтология, отражающая структуру игры, идентифицирует важные структурные элементы игры и отношения между ними. Их описание представляет иерархически организованную систему игровых понятий (терминов), составляющих таксономию игры. Онтология отражает семантические связи между понятиями предметной области и может быть основой для исследования, анализа и понимания взаимосвязей между различными игровыми элементами и процессом принятия решений игроком, а также позволяющая генерировать выводы о намерениях и целях игрока. По сути, онтология обеспечивает семантическую организацию хранения данных, извлеченных из игровых логов, дополнительных данных, полученных на основе стандартных методов DataMining и комплекса правил вывода заключений о демонстрируемом поведении в игре.

Модуль оценки параметров поведения игрока реализует найденные закономерности и шаблоны поведения, демонстрируемые игроком на протяжении игровых сессий. Выявление шаблонов поведения игрока на основе ассоциативных правил, выявляющих зависимостей между связанными событиями (из события X следует событие Y), позволяют исследовать зависимости в игровых логах. Это обеспечит выявление и учет персональных предпочтений игрока, может использоваться в политике удержания игроков в игре.

Модуль типизации игрового поведения групп игроков в онлайн-игре. Для выявления закономерностей игрового поведения и классификации его типов на основе собранной информации могут быть использованы классические методы DataMining.

Сегментирование и построение профилей игроков возможно с использованием кластеризации и деревьев принятия решений. Сегментация игроков позволяет узнать, какие именно сегменты являются наиболее активными, какие приносят наибольшую прибыль в игре, выделить характерные для них признаки. В качестве инструментов анализа использованы различные инструменты аналитики (библиотеки Python).

Заключение

В статье был проведен анализ возможных подходов к исследованию процесса принятия решений участниками онлайн игры и его семантической интерпретации. Разработана архитектура системы Game Analytics, поддерживающей смысловой анализ поведения игроков.

Предложен подход к анализу целенаправленности игрового поведения на основе отслеживания циклов деятельности при исследовании игровых логов. Система распознавания целей игрока использует результаты парсинга логов. Из общего числа логов выделяются последовательности для классификации игровых ситуаций, в которых игроки для достижения своих целей выбирают действия (средства достижения целей) с учетом своего понимания эффективности действия на уровне тактики и стратегии в игре.

Список литературы

1. Гаршин Т. С. Обзор технологий Game-аналитики для исследования поведения игроков в on-line играх / Т. С. Гаршин, А. Ю. Иванков; отв. ред. Д. Н. Борисов // 19-ая Международная конференция «Информатика: проблемы, методология, технологии», Воронежский государственный университет. – Воронеж : Издательский дом ВГУ, 2019. – С. 1441-1446.

2. Гаршин Т. С. Классификация типов игрового поведения участников on-line игр / Т. С. Гаршин, А. Ю. Иванков; отв. ред. Д. Н. Борисов // Сборник студенческих научных работ факультета компьютерных наук ВГУ. Выпуск 13 : в 2 ч. – Вып. 13. – Ч.1. – Воронеж : Издательский дом ВГУ , 2019. – С.46-52.

3. Djuric B. O. Specific owl-based rpg ontology / B. O. Djuric and M. Konecki // Central European Conference on Information and Intelligent Systems (26th International Conference), 2015. – P. 27-66.

4. Towards an Ontological Language for Game Analysis / Zagal Jose P., Mateas Michael [et al.] // Proceedings of the Digital Interactive Games Research Association Conference (DiGRA), Tokyo, Japan, 2007. – P. 516-523.

5. The Video Game Ontology / J. Parkkila [et al.]. – Technical report, 2014. – P. 155-162.

Повышение эффективности управления текущей деятельностью компании

Ю. В. Гостева

Студент бакалавриата

И. В. Абрамов

Доцент

Введение

Обеспечение эффективности управления является целью любой компании, так как именно это является залогом успешного функционирования и развития ее деятельности. При этом важно правильно оценивать работу людских ресурсов, знать их возможности, принимать оптимальные управленческие решения, для предотвращения появления рисков из-за человеческого фактора. Грамотный выбор сотрудника предоставляет возможность не только сократить сроки исполнения проекта, но и повысить качество выполнения работ и эффективность компании в целом.

Проблема заключается в том, что для достижения наилучшего результата и сокращения времени, выбор сотрудников для конкретного проекта не всегда является оптимальным, так как лицо, принимающее решение, зачастую основывается на субъективной оценке, а не на методе, который мог бы объективно оценить возможности оптимального распределения людских ресурсов [1].

В данной работе рассматривается задача распределения работников на проект с использованием метода анализа иерархий.

Задача включает в себя:

1. Анализ деятельности сотрудников компании и выявление основных критериев;

2. Определение степени значимости каждого критерия и их попарное сравнение;
3. Разработку системы оценки сотрудников по выделенным критериям;
4. Построение обобщенного критерия соответствия кандидатов требованиям организации;
5. Принятие решений на основе результата.

1. Выявление основных критериев и определение степени их значимости.

Постановка задачи: в коммерческой компании имеется проект, на который нужно назначить пять сотрудников. Требуется сформировать оптимальную рабочую группу.

Для использования метода анализа иерархий необходимо определить базисные критерии, которыми обладают все сотрудники. В табл. 1 составлен список, который позволяет учесть преимущества и недостатки претендентов в рабочую группу.

Таблица 1

*Критерии для цели
«Оптимальный выбор сотрудников на проект»*

№	Наименование критерия	Описание
1	Загруженность сотрудника	Занятость работника в других проектах.
2	Образование	Уровень образования.
3	Опыт работы	Теоретические знания сотрудника и умение применить их в работе.
4	Количество завершенных проектов	Сумма выполненных проектов
5	Организаторские способности	Профессиональная компетентность в сочетании с лидерскими и педагогическими знаниями.
6	Мобильность	Возможность выездных командировок, возможность изменения графика работы.

Далее необходимо сравнить их в отношении к цели. Определять степень значимости будем с использованием девятибалльной шкалы, которая предложена математиком Т. Саати [2]. При сравнении необходимо следовать данным правилам:

1. Если сравниваемые критерии имеют равную значимость, то важность равна 1;
2. Если один критерий имеет умеренное превосходство над другим, то важность равна 3;
3. Если наблюдается существенное превосходство критерия, то важность равна 5;
4. Если критерий имеет значительное превосходство, то важность равна 7;
5. Если критерий имеет очень сильное превосходство, то важность равна 9;
6. 2, 4, 6, 8 имеют промежуточную значимость между соседними суждениями для «сглаживания» превосходства.

Результаты оценки в табл. 2 являются обратно симметричной матрицей с единицами на главной диагонали. Для того чтобы оценить логичность суждений необходимо воспользоваться формулами [2].

Таблица 2

Матрица попарных сравнений и определения степени значимости

	Загружен ность	Обр зова ние	Опыт работы	Кол-во проектов	Личные качества	Мобильн ость	W_i
Загруженность	1	3	1/2	1/3	5	6	0,190
Образование	1/3	1	1/3	1/3	1/2	4	0,077
Опыт работы	2	3	1	4	6	9	0,400
Кол-во проектов	3	3	1/4	1	3	5	0,228
Личные качества	1/5	2	1/6	1/3	1	3	0,075
Мобильность	1/6	1/4	1/9	1/5	1/3	1	0,030

Максимальное собственное число:

$$\lambda_{\max} = \sum_{i=1}^n v_i \cdot w_i, \quad (1)$$

где $\sum_{i=1}^n v_i$ – сумма столбца суждений, w_i – нормализованный собственный вектор.

Индекс согласованности:

$$ИС = \frac{\lambda_{\max} - n}{n - 1}, \quad (2)$$

где n – размерность матрицы парных сравнений.

Отношение согласованности:

$$ОС = \frac{ИС}{M(ИС)}, \quad (3)$$

где $M(ИС)$ – число случайной согласованности, для матрицы 6 порядка равно 1,24 [3].

Результаты подсчета, выполненные средствами MS Excel: $\lambda_{\max} = 6,599$, $ИС = 0,120$, $ОС = 0,097$.

Как мы видим, отношение согласованности меньше 10%. Это говорит нам о том, что при заполнении матрицы не произошло нарушения логичности суждений. Наибольшей степенью значимости обладает критерий опыт работы.

2. Подбор оптимальной рабочей группы.

Альтернативами в данном исследовании являются сотрудники компании. Необходимо оценить их по выделенным критериям. Для этого критерии были разделены на уровни, каждому из которых поставлен в соответствие отрезок на шкале от 0 до 1. Например, для критерия «Образование» шкала будет выглядеть следующим образом:

1. Высшее специальное + знание нескольких языков [0,9; 1);
2. Высшее специальное + знание одного языка [0,8; 0,9);
3. Высшее по специальности [0,6; 0,8);
4. Высшее [0,4; 0,6);
5. Неоконченное высшее [0,2; 0,4);
6. Среднее специальное образование [0,1; 0,2).

Возьмем сотрудника «Колесников Вячеслав Юрьевич», на данный момент он занят в 3 проектах (0,4), имеет высшее образование по специальности и владеет английским языком (0,8), работает в компании 5 лет (0,6), за это время завершил 24 проекта (0,7), его организаторские качества можно оценить, как специалист, у которого более выражена профессиональная компетенция (0,4), является мобильным сотрудником с возможностью командировок несколько раз в полгода (0,3).

Значение критерия соответствия вычислим по формуле [4]:

$$O_j = \sum_{i=1}^n w_i \cdot d_{ij}, \quad (4)$$

где w_i – степень значимости i -ого критерия, а d_{ij} оценка j -го кандидата по i -му критерию.

Для первого кандидата $O_j = 0,5766$. Результаты расчетов критерии соответствия для всех кандидатов отражены в табл. 3.

Таблица 3

Итоговая оценка сотрудников методом анализа иерархий

Сотрудник	Значение критерия соответствия.	Рейтинг
Авдеев Константин Олегович	0,7423	1
Андреева Екатерина Анатольевна	0,3521	7
Барыкина Мария Юрьевна	0,2987	9
Ковалева Анастасия Дмитриевна	0,1932	10
Колесников Вячеслав Юрьевич	0,5766	4
Колпаков Дмитрий Владимирович	0,6305	2
Красильников Василий Иванович	0,5024	5
Мандрыкин Станислав Борисович	0,6005	3
Пантелеев Александр Николаевич	0,1743	11
Сайко Владислав Эдуардович	0,4987	6
Хрусталева Елизавета Антоновна	0,3048	8

В результате применения метода анализа иерархий, мы можем сделать вывод, что оптимальная рабочая группа на проект будет состоять из кандидатов с рейтингом с 1 по 5.

Заключение

В рамках данной работы был применен метод анализа иерархии для определения состава командной группы на работы проекта.

Использование данного подхода позволит увеличить эффективность управления компанией за счёт математических методов, которые учитывают влияние всевозможных критериев на принятие решений. Выбор сотрудников на проект является важным решением, так как оно напрямую влияет на работоспособность операционного уровня управления компанией. Данное исследование позволяет лицу, принимающему решение, уменьшить временные затраты на формирование оптимальных проектных команд, нацеленных на достижение наилучшего результата.

В качестве дальнейшего направления работы можно рассмотреть возможность применения данного метода для найма персонала, подбора кандидата на повышение в должности или для иных задач.

Список использованной литературы

1. Карпов, В. А. Проблемы принятия решения в трудовой деятельности / В. А. Карпов // Управление персоналом. – 1999. – Т. 14. – № 3. – С. 3-14.
2. Саати, Т. Л. Принятие решений. Метод анализа иерархий / Т. Л. Саати. – М. : Радио и связь, 1993. – 167 с.
3. Горелик, В. А. Теория принятия решений : учебное пособие для магистрантов / В. А. Горелик. – М. : МПГУ, 2016. – 152 с.
4. Саати, Т. Л. Принятие решений при зависимостях и обратных связях. Аналитические сети / Т. Л. Саати. – М. : Книжный дом «Либроком», 2009. – 360 с.

Разработка модулей сетевого ресурса и конфигурации сети для web-интерфейса SOHO-сервера

М. Б. Дремин

Студент бакалавриата

А. А. Головкин

Ассистент

Введение

В настоящее время неотъемлемой составляющей современных квартир и жилых домов, конкурентоспособных предприятий и компаний является внутренняя ИТ–система, структура которой определяется в зависимости от требуемых задач. Развитие и повсеместное использование таких решений, как оборудование и ресурсы класса SOHO, обусловлено постоянно растущим числом пользователей и сервисов, а также необходимостью организации передачи и обмена информации, настройки сетевой конфигурации и подключения физических и виртуальных устройств.

SOHO–серверы (Small Office Home Office – малый офис, домашний офис) входят в состав одноименного типа сети и предназначены для разностороннего обслуживания небольшого числа рабочих станций при условии использования дома или в офисе [1].

Наиболее подходящим способом настройки компонентов сервера такого класса считается задействие специального web–интерфейса, в котором все необходимые операции проводятся при задействовании его

визуальной составляющей на странице в Интернет-браузере. Особенностью существующих решений по настройке конфигурации маломасштабных сетей зачастую является их функциональная избыточность, что может затруднять их полноценное использование в условиях дома или офиса.

Представленная в статье работа посвящена разработке и программной реализации модулей для web-интерфейса, который будет способен обеспечить интуитивно понятные настройки и последующее обслуживание нескольких компонентов, необходимых для деятельности в сети рабочих станций и устройств: модуля общего сетевого ресурса и модуля конфигурации сети.

1. Структура сети класса SOHO

Стандартная схема сети такого класса, как правило, представляет собой физическую иерархию компонентов, также именуемую топологией. В наиболее распространенном варианте SOHO-конфигурации, представленном на рис. 1, сервером является пользовательская рабочая станция, снабженная необходимым ПО для контроля над организуемой сетью и предоставляющие различные услуги. Актуальные версии маршрутизаторов совмещают в себе роли коммутатора и точки беспроводного доступа, что оказывает положительное влияние на процесс обслуживания комплектующих.

2. Реализация модуля общего сетевого ресурса

Рассмотрение данного аспекта следует начать с разъяснений относительно используемых при разработке технологий. В работе над модулем использован инструмент SMB (Server Message Block) – сетевой протокол для общего доступа к файлам и директориям, построенный по принципу «клиент-сервер», который предоставляет клиентским компьютерам простой способ для чтения и записи файлов, а также запроса служб у серверных программ в различных типах сетевого окружения – и в частности его свободно распространяемая реализация для дистрибутивов ОС Linux – Samba [2]. Протокол используется на уровнях Application (Приложения) и Presentation (Представления) сетевой модели OSI и зависит от транспортных протоколов, таких как TCP/UDP, а также дублирует функционал сервера Доменов Active Directory (технология Windows).

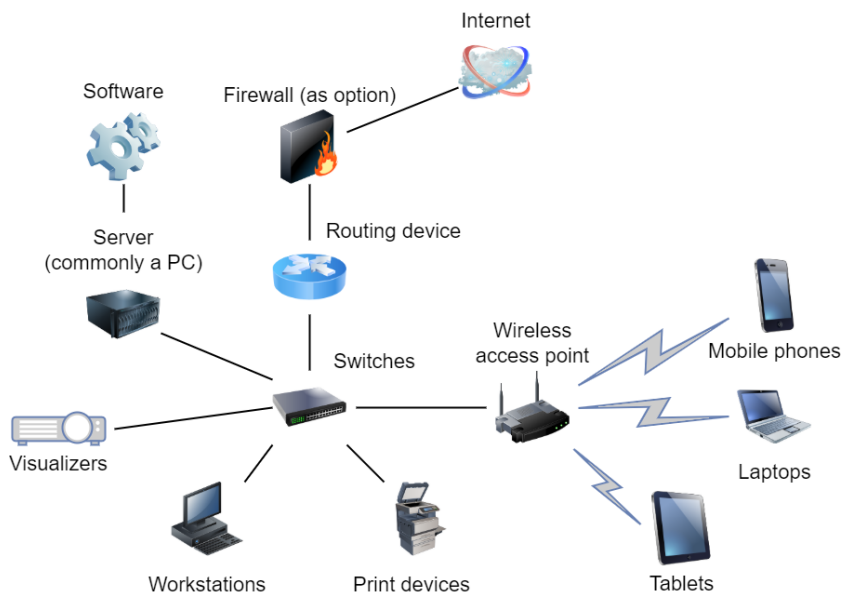


Рис. 1. Пример построения сети класса SOHO

Пример установки соединения по транспортному протоколу, отображающий сущность SMB, обозначенную как «request—response» («запрос—отклик»), представлен на рис. 2.

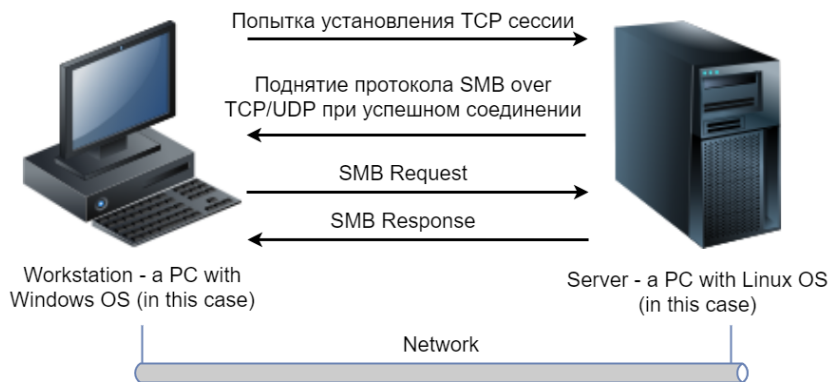


Рис. 2. Принцип работы инструмента SMB

Создание общего ресурса происходит с присвоением статуса разрешения/запрета на запись и/или чтение, при задании параметров

определяется владелец для доступа через процедуру аутентификации. Интерфейс модуля представлен на рис. 3.

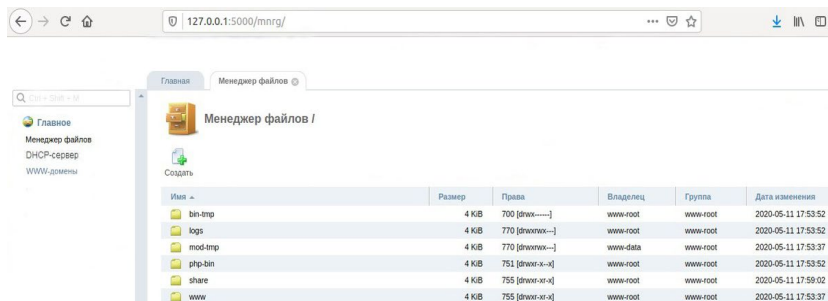


Рис. 3. Интерфейс модуля общего сетевого доступа

3. Реализация модуля конфигурации сети

В этом разделе статьи будет рассматриваться такой вариант настройки параметров, как присвоение уникальных IP—адресов компьютерам—клиентам сети с помощью «протокола динамического конфигурирования хостов» – DHCP [3]. Механизм также построен по принципу «клиент—сервер», используется на Прикладном уровне модели OSI и включает в себя два компонента: протокол предоставления конфигурационных параметров и механизм назначения сетевых адресов.

Динамическое присвоение адресов является оптимальной схемой для клиентов, подключаемых к сети временно или совместно использующих один и тот же набор IP—адресов и не нуждающихся в них постоянно. В данной процедуре просматривается типичная схема взаимодействия клиента и сервера DHCP путем обмена сообщениями определенного формата, что демонстрирует диаграмма на рис. 4.

Стоит отметить, что клиент может решить отказаться от сетевого адреса путем отправки серверу сообщения DHCPRELEASE, данная процедура аннулирует оставшееся время действия адреса. Пункт «Источник IP—адреса» подразумевает возможность автоматического получения адресов в сети для облегчения процесса настройки. Интерфейс модуля конфигурации сети представлен на рис. 5.

В реализации задействованы среда программирования Python [4, 5] в связке с фреймворком Flask для создания web—продуктов. В качестве web—сервера был выбран Nginx, работа выполнена на базе дистрибутива ОС Linux Ubuntu 18.04 LTS, для его использования как дополнительной системы привлечена виртуальная машина Oracle VM VirtualBox.

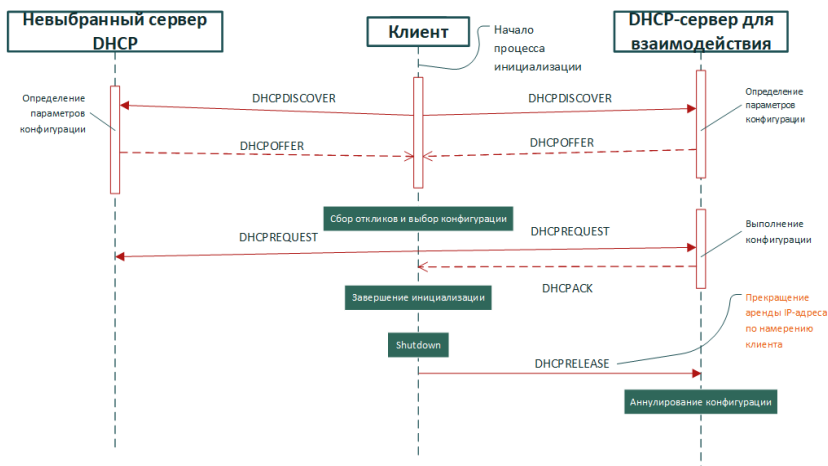


Рис. 4. Диаграмма последовательностей взаимодействия

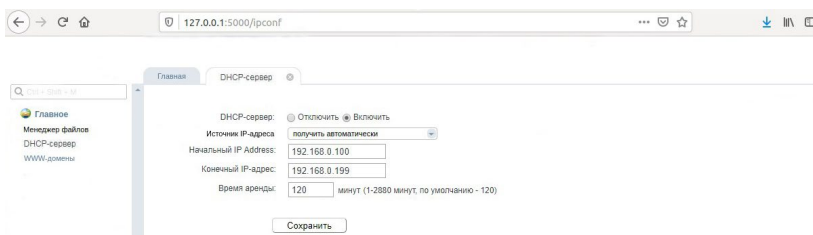


Рис. 5. Интерфейс модуля конфигурации сети

Заключение

В данной статье рассмотрены варианты разработки и реализации модулей для web-интерфейса, который выполняет определенные задачи по конфигурации сервера класса SOHO. Приведено описание организации сессии по протоколу SMB для осуществления сетевой коммуникации и процесса взаимодействия между DHCP сервером и клиентом для получения последним IP-адреса. Предложенные алгоритмы были реализованы в виде прототипа в среде web-разработки. Использование данного решения позволяет организовать общий файловый ресурс и настроить IP-конфигурацию устройств сети с использованием интерфейса, а в дальнейшем предполагается улучшить текущие наработки и расширить список охватываемых параметров.

Список литературы

1. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы : учебник / В. Олифер, Н. Олифер. – 6-е изд. – М. : Питер, 2016. – 996 с.
2. Электронная библиотека университета ИТМО [Электронный ресурс]: информационный портал. – Режим доступа : <https://neerc.ifmo.ru>
3. Vaughan National Library [Электронный ресурс] : электронный науч. журн. – Режим доступа : <https://ru.bmstu.wiki>
4. Бэрри, П. Изучаем программирование на Python / П. Бэрри. – 2-е изд., перераб. и доп. – М. : Эксмо, 2017. – 624 с.
5. Unix и Linux. Руководство системного администратора : пособие / Эви Немет [и др.]. – 4-е изд. – М. : Вильямс, 2012. – 1312 с.

Модель системы хранения документов об образовании с использованием технологии блокчейн

А. Д. Дроботенко

Студент бакалавриата

А. А. Головкин

Ассистент

Введение

Слово «блокчейн» зачастую ассоциируют с криптовалютой. Однако эта технология имеет гораздо более широкие возможности применения. Блокчейн – децентрализованная база данных, которая хранит информацию всех участников сети в виде «цепочки блоков» (именно так это слово переводится с английского). Технология обеспечивает сохранность данных, их неизменность и безопасность.

На сегодняшний день распределенная база данных блокчейн все больше интегрируется в системы хранения и контроля документов. Первыми, кто официально начал использовать блокчейн для хранения своих дипломов и сертификатов стал Университет Никосии [1]. Он также стал первым университетом, который принимает в качестве оплаты криптовалюту Bitcoin. Так же более 100 выпускников Массачусетского Технологического Института уже получили цифровые дипломы на базе этой технологии. Японская компания Sony, создавшая

сервис Sony Global Education, использует блокчейн при выдаче сертификатов с 2017 года.

В данной статье рассмотрена модель системы хранения документов об образовании с использованием технологии блокчейн [2].

1. Блокчейн как распределенная система

Блокчейн – непрерывная последовательная цепочка блоков с информацией, выстроенная по определенным правилам с целью обеспечения доверия и надежности транзакций за счет технологии распределенного реестра.

Это хронологическая база данных, то есть такая база данных, в которой время, когда была сделана запись, неотделимо от самих данных, что делает ее некоммутативной. Основными структурными элементами являются блоки. В них хранятся данные вместе со служебной информацией: хэш текущего блока, хэш предыдущего блока, временная метка. Новый блок добавляется к предыдущему, образуя цепь. Блоки соединены в односвязный список. Каждый участник представлен узлом, который хранит весь список данных и контактирует с другими узлами. Узлы могут добавлять новые записи в конец списка.

Особую роль в формировании блоков играют хэш-функции [3]. Использование данных функций обусловлено следующими их преимуществами:

1. Устойчивость к коллизиям – невозможно получить два одинаковых хэша при разных начальных данных.
2. Эффект Лавины – небольшое изменение входных данных приводит к сильному изменению хэша.
3. Необратимость хэша – сложность получения исходных данных из итогового хэша.

2. Методы защиты в системе

Деятельность узлов-операторов ограничена определенными правилами. Они добавляют новые блоки в соответствии с определенными принципами. Благодаря этим принципам обеспечивается безопасность блокчейна и его децентрализация. Консенсус – единое мнение о том, какую версию блокчейна считать верной.

Для частных блокчейнов в качестве механизма достижения консенсуса удобно взять алгоритм Proof-of-authority. Модель PoA основана на ограниченном количестве валидаторов блока (одобренных узлов), и именно это делает систему масштабируемой. Модель Proof-of-authority позволяет компаниям сохранять свою конфиденциальность, пользуясь преимуществами технологии блокчейн.

3. Проектирование системы

Разрабатываемая система хранения документов об образовании должна удовлетворять следующим требованиям:

1. Хранение заданных документов в цепочке блокчейн.
2. Возможность просмотра необходимых документов при обращении.
3. Упрощение процесса выдачи документов об образовании для учебных заведений.

В данной концепции будет использоваться эксклюзивный блокчейн. Эксклюзивные блокчейны обеспечивают больший контроль над системой со стороны учебного заведения (большая гибкость и функциональность).

Валидацию данных осуществляют узлы-операторы. У каждого обработчика транзакций есть пара ключей – закрытый и открытый. Создатели блоков известны и определяются по цифровой подписи блока [4].

Алгоритм работы проектируемой системы отражён на рис. 1 и 2.

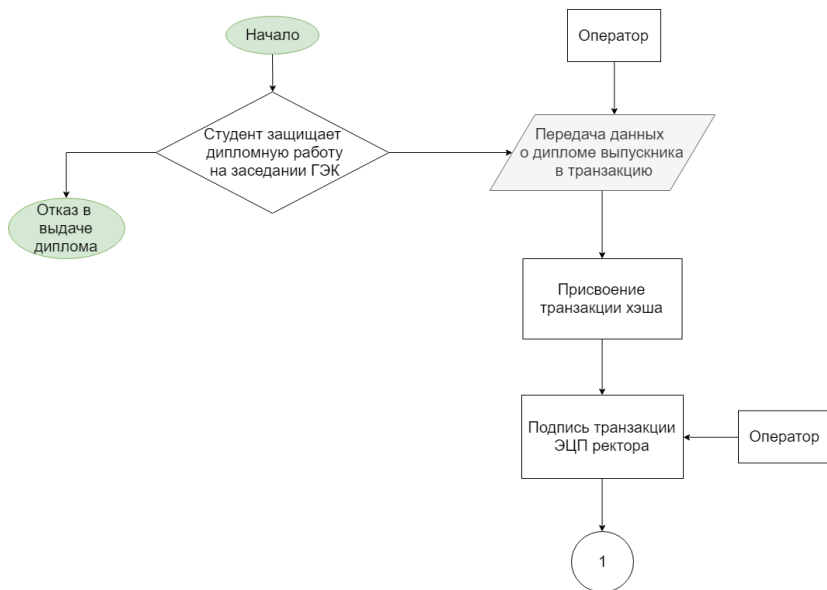


Рис. 1. Алгоритм работы системы хранения документов об образовании



Рис. 2. Алгоритм работы системы хранения документов об образовании. Окончание

Выпускник защищает диплом в присутствии ГЭК. После этого уполномоченное лицо (оператор) создает цифровой диплом в виде транзакции и передает данные о дипломе и его хэш в блокчейн и заверяет все это ЭЦП ВУЗа. Одна транзакция соответствует одному выпускнику. Цифровой диплом включает в себя следующие данные:

- Наименование ВУЗа;
- ФИО выпускника;
- Серия и номер диплома;
- Электронная копия диплома (скан);
- Дата выдачи диплома.

4. Реализация системы

Для реализации модели системы хранения документов об образовании на базе технологии блокчейн был выбран язык программирования Java совместно с использованием фреймворка Spring

– универсальный фреймворк с открытым исходным кодом для Java-платформы. Для создания пользовательского интерфейса используется JavaScript-фреймворк Vue.js.

Для демонстрации работы системы были созданы три типа пользователей с уникальными функциями:

- Оператор – пользователь, который может добавлять документы в блокчейн.

- Рядовой пользователь – основная функция – просмотр блока с данными по хэшу.

- Хакер – пользователь, который осуществляет несанкционированный доступ к блокчейну с целью подмены блока.

- Каждый блок содержит хэш текущего блока, хэш предыдущего блока, временную метку и поле данные, где хранится вся информация о дипломе.

Общий вид интерфейса реализованной системы представлен на рис. 3.

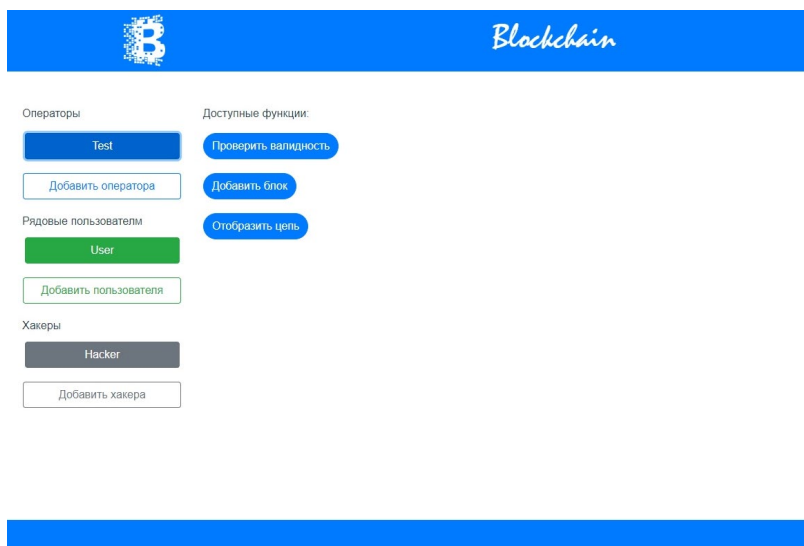


Рис. 3. Общий вид интерфейса системы

Интерфейс добавления документа в систему представлен на рис. 4.

Добавить диплом

ФИО Студента

Введите ФИО Студента

Наименование ВУЗа

Данные диплома

Выбрать... Обзор

Серия

Номер

Дата выдачи

Отмена Добавить

Рис. 4. Интерфейс добавления документа

Заключение

Данная статья посвящена проектированию и разработке модели хранения документов об образовании с использованием технологии блокчейн. Была реализована модель, которая наглядно демонстрирует технологическую возможность применения данной технологии в сфере образования:

1. Прозрачность системы выдачи дипломов выпускникам учебных заведений. Блок блокчейна невозможно изменить или удалить цепочки блоков без последствий. При этом любой участник образовательного процесса, имеющий свой публичный ключ, в любой момент может подтвердить «истинность» диплома.

2. Общее сокращение бумажного документооборота, перевод классических процессов в сферу информационного пространства.

Список литературы

1. The University of Nicosia (UNIC) is, by most measures, the leading university globally in the digital currency and blockchain field. [Электронный ресурс] / Digital Currency is an Internet-based medium of exchange with properties similar to physical currency. – Режим доступа : <https://digitalcurrency.unic.ac.cy/>

2. Grech, A. Blockchain in Education [Электронный ресурс]. – Режим доступа : http://publications.jrc.ec.europa.eu/repository/bitstream/JRC108255/jrc108255_blockchain_in_education%281%29.pdf

3. Хэш-функции [Электронный ресурс] / Википедия – свободная энциклопедия. – Режим доступа : <https://ru.wikipedia.org/wiki/Хеш-функция>

4. Электронная цифровая подпись [Электронный ресурс] / КриптоПро. – Режим доступа : <https://www.cryptopro.ru/products/cades>

Система учёта успеваемости на факультете 2.0

И. А. Еремин

Студент бакалавриата

А. И. Чекмарев

Ассистент

Введение

В современном мире образовательные учреждения часто сталкиваются с проблемой излишнего складирования ненужной документации. Это заметно сказывается на процессе обучения и эффективности работы сотрудников университета.

Поэтому целью моей работы является создание такого сервиса, который позволит пользователю эффективно оптимизировать процесс обучения: вводить и просматривать результаты экзаменов и аттестаций студентов, также хранить данные об обучающихся, учебном плане и преподавателях.

1. Постановка задачи

Цель: разработать приложение для учета успеваемости студентов для факультета компьютерных наук.

Сфера использования: образовательный процесс факультета.

Требования:

1. Логика выставления оценок за рубежные аттестации и вычисление итоговых оценок должно полностью соответствовать положению «о текущей и промежуточной аттестации знаний, умений и навыков студентов в балльно-рейтинговой форме на факультете компьютерных наук Воронежского государственного университета» от 11.06.2019 г. [1].

2. Авторизация пользователей осуществляется при помощи существующей на факультете Active Directory.

3. Выделить три роли пользователей:

– Студент — может только просматривать результаты аттестаций и экзаменов.

– Преподаватель — может просматривать результаты аттестаций и экзаменов, а также выставлять оценки для своих групп.

– Администратор (сотрудник деканата) — выполняет формирование студенческих групп, учебного плана, а также выставлять оценки за аттестации и экзамены, которые преподаватель предоставил в деканат на бумажной ведомости.

4. Возможность поиска информации об экзаменах, аттестациях, студентах, студенческих группах и преподавателях.

5. В данном приложении не будут учитываться повторные зачёты и экзамены (пересдачи за предыдущий семестр), поскольку данные функции уже реализованы в другом существующем приложении.

6. Применяется только для бакалавриата и специалитета.

7. Реализовать в виде веб-приложения.

Задачи:

– Провести анализ требований к разрабатываемой системе.

– Спроектировать приложение.

– Реализовать приложение, удовлетворяющее указанным требованиям, и описать процесс разработки и итоговый результат.

2. Определение бизнес-процессов

В качестве бизнес-процессов можно выделить следующие этапы:

1. Зачисление студента.

2. Создание студенческих групп.

3. Создание учебного плана для каждой группы и закрепление преподавателя за каждой единицей учебного плана.

4. Выставление оценок за аттестации и экзамены.

5. Перевод студентов на следующий семестр или оформление выпуска студентов

6. Дополнительно необходимо предусмотреть возможность отчисления студента или его уход в академический отпуск.

3. Средства реализации

В качестве средств реализации был выбран язык Python версии 3.8 [2] со следующими модулями:

1. SQLAlchemy — библиотека для работы с реляционными СУБД с применением технологии ORM (объектно-реляционное отображение).

Служит для синхронизации объектов Python и записей реляционной базы данных.

2. Flask — фреймворк для создания веб-приложений. Со следующими расширениями [3]:

- Flask-Login — расширение Flask для авторизации;

- Flask-SQLAlchemy — расширение для упрощённого доступа к SQLAlchemy через Flask.

3. WTForms — библиотека для упрощённой обработки html-форм, с возможным добавлением валидаторов (условий допустимых значений для полей формы).

4. WTForms-Alchemy — библиотека для связи объектов WTForms и объектов SQLAlchemy.

5. ldap3 — библиотека для работы по протоколу LDAP для связи с Active Directory.

6. mysql-connector — библиотека для подключения с СУБД MySQL.

В качестве реляционной СУБД выбрана MySQL [4].

4. Схема данных

В результате анализа предметной области выделены следующие сущности: студент, преподаватель, студенческая группа, предмет (дисциплина), единица учебного плана, оценка за аттестацию и экзамен. Дополнительно можно выделить не связанную ни с какими объектами сущность — администратор.

Под студенческой группой в данной схеме понимается студенческая группа в определённом семестре. При переводе студентов на следующий семестр, будет создана новая запись группы на семестр выше и к которой привязываются студенты, а у старой группы устанавливается снимается атрибут «является действующий».

Студент может иметь следующие состояния: учится, успешно закончил обучение, отчислен, в академическом отпуске. Если студент в состоянии учится, то обязательно должна быть ссылка на студенческую группу, а в остальных состояниях ссылки быть не должно. Если студент успешно закончил обучение, то должно быть запись о годе выпуска, а если отчислен или в академическом отпуске, то указан год перехода в данное состояние.

Базовым объектом в схеме является единица учебного плана. Единица учебного плана ссылается на преподавателя, студенческую группу, предмет (из списка), тип отчётности (зачёт, экзамен, зачёт с оценкой), преподавателя.

Для объектов студент, преподаватель, администратор добавлен атрибут учётное имя. Оно необходимо для авторизации и совпадает с учётным именем в существующей Active Directory факультета

В первой версии приложения все поля аттестационной ведомости могли свободно редактировать любое поле преподаватели и секретарь, что не соответствовало положению о БРС на ФКН.

По сравнению с предыдущей версией приложения были добавлены новые таблицы.

Таблица curriculum_unit:

+curriculum_unit_status – поле состояние аттестационной ведомости. Может принимать следующие значения:

– 'att_1' – Выставление оценок за 1-ю аттестацию (данное значение по умолчанию устанавливается при создании единицы учебного плана);

– 'att_2' – Выставление оценок за 2-ю аттестацию;

– 'att_3' – Выставление оценок за 3-ю аттестацию;

– 'exam' – Выставление оценок за экзамен (для типа отчётности – «экзамен») и дополнительных баллов (для типа отчётности – «экзамен» или «зачёт с оценкой»);

'close' – Ведомость закрыта - редактирование данных ведомости не допускается.

Таблица curriculum_unit_status_hist. История изменения значения поля curriculum_unit_status в таблице curriculum_unit

curriculum_unit_id – Ссылка на первичный ключ таблицы curriculum_unit(поле curriculum_unit_id);

stime – Время начала действия записи;

etime – Время окончания действия записи (если null, то действует на текущий момент);

admin_user_id – Ссылка на пользователя «администратор», выполнившего изменение;

curriculum_unit_status – поле состояние аттестационной ведомости. Аналогично полю в таблице curriculum_unit.

Таблица att_mark_hist. История изменения значения полей в таблице att_mark.

– att_mark_id – Ссылка на первичный ключ таблицы att_mark(поле att_mark_id);

– stime – Время начала действия записи;

– etime – Время окончания действия записи (если null, то действует на текущий момент);

– admin_user_id – Ссылка на пользователя «администратор», выполнившего изменение;

- teacher_id - Ссылка на пользователя «преподаватель», выполнившего изменение;
- Поля 'att_mark_1', 'att_mark_2', 'att_mark_3', 'att_mark_exam', 'att_mark_append_ball' - Аналогичны соответствующим полям в таблице att_mark.

Примечание в таблице att_mark_hist должно быть заполнено одно из полей admin_user_id или teacher_id.

Итоговая схема данных представлена на рисунке.

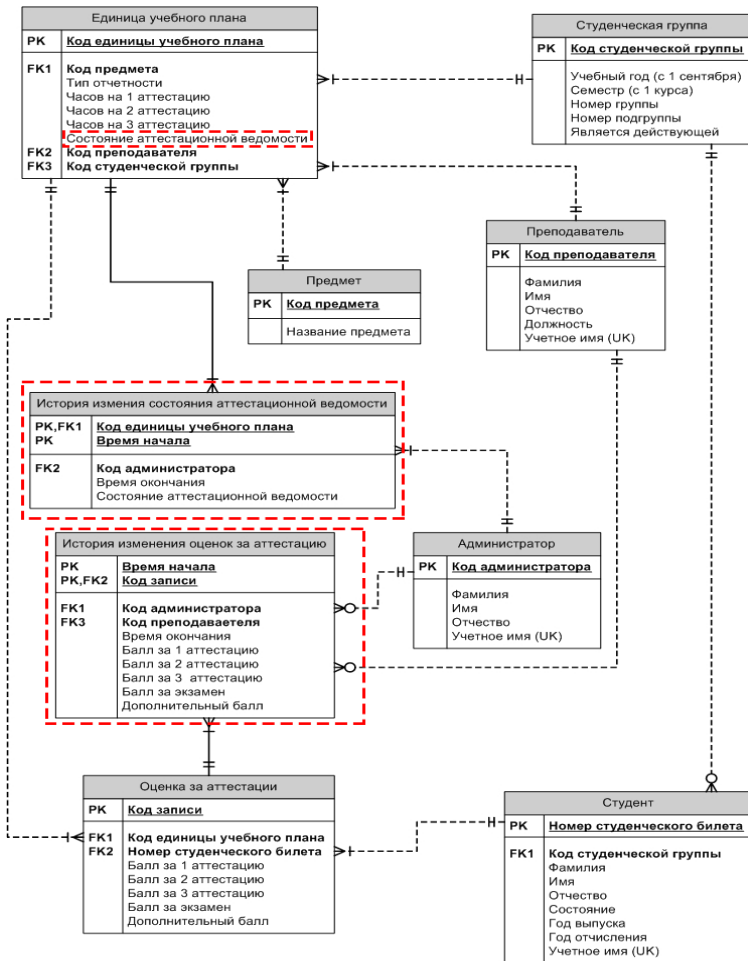


Рисунок. Обновленная схема данных приложения

Указанная схема данных позволяет сохранять в полной мере исторические данные об аттестациях и экзаменах без потерь при переводе студентов на следующий семестр, при выпуске и отчисления студента, при разных учебных планах для каждого года набора студентов. В данной схеме допустим переход студента из одной группы в другую при переходе на новый семестр. При формировании аттестационной ведомости для указанной единицы учебного плана создаются записи для студентов соответствующей группы. А для просмотра результатов аттестаций для студента по всем семестрам просматриваются все оценки за аттестации, а через эти записи по коду единицы учебного плана можно узнать студенческую группу.

Правила заполнения аттестационных ведомостей

В соответствии с вышеуказанным положением было разработано правило заполнения полей аттестационной ведомости в зависимости от текущего этапа, которое указано в таблице.

Таблица

Правило заполнения полей аттестационной ведомости

Состояние ведомости	1 аттестация	2 аттестация	3 аттестация	Оценка за экзамен*	Доп. балл**
1 аттестация	Заполняется	-	-	-	-
2 аттестация	заполнено ***	заполняется	-	-	-
3 аттестация	заполнено ***	заполнено ***	заполняется	-	-
подведение итогов	заполнено** *	заполнено ***	заполнено ***	заполняется	заполняется
закрыта	заполнено	заполнено	заполнено	заполнено	заполнено

- — заполнять нельзя

заполнено — Полное должно быть заполнено при переходе на данный этап. Изменять нельзя

заполнено — Полное должно быть заполнено при переходе на данный этап. Можно изменить, если оценка не положительная (менее 25 баллов).

* Оценка за экзамен выставляется только для типа отчётности «Экзамен»

** Дополнительный балл выставляется только для типа отчётности «Экзамен» и «Зачёт с оценкой».

*** Данные поля может редактировать секретарь. Предусмотрено для исправления данных.

5. Функциональные модули

Для удобства работы с данным шаблоном написан ряд макросов. В данном приложении выделены следующие функциональные модули:

1. Карточка студента.
2. Карточка студенческой группы.
3. Редактор списка преподавателей.
4. Редактор списка(справочника) предметов (дисциплин).
5. Редактор учебного плана для группы.
6. Редактор аттестационной ведомости.

Модуль переноса студентов на следующий семестр.

Отдельно выделим модуль переноса студентов на следующий семестр. Действие применимо к отдельной группе. Перенос группы на следующий семестр возможен когда для всех единиц учебного плана были заполнены ведомости. При переносе на следующий семестра создаётся копия записи о студенческой группе на семестр больше, у старой группы снимается флаг «является действующей», все студенты старой группы привязываются к новой. Возможно в данном модуле оформление выпуска студентов. При оформлении выпуска, тоже у группы снимается флаг «является действующей», а у студентов переходят в состояние «успешно закончил обучение» и устанавливается текущий год выпуска.

6. Системные требования

Для развёртывания приложения необходим веб-сервер Apache2 или nginx с модулем WSGI. На сервере необходим интерпретатор Python версии не ниже 3.8, с модулями указанными в п. 3.

Также необходима СУБД MySQL версии не ниже 5.8.

Для клиентской части необходим любой современный веб-браузер.

Заключение

В результате проделанной работы было разработано веб-приложение, полностью удовлетворяющее указанным требованиям. Успешно функционирует аудит изменения данных. Дополнительно планируется добавить в приложение возможность для печати аттестационных ведомостей.

Планируется внедрение в следующем учебном году системы учёта успеваемости на факультете компьютерных наук Воронежского государственного университета.

Список литературы

1. Положение о текущей и промежуточной аттестации знаний, умений и навыков студентов в балльно-рейтинговой форме на факультете компьютерных наук Воронежского государственного университета от 11.06.2019 г. / П ВГУ 2.1.04.16 – 2019.

2. Python 3.8 documentation [Электронный ресурс] : база данных. – Режим доступа : <https://docs.python.org/3.8/>

3. Grinberg, M. Flask Web Development / M. Grinberg. – First Edition. – Cambridge : O'Reilly, 2014. – 238 p.

4. MySQL documentation [Электронный ресурс] : база данных. – Режим доступа : <https://dev.mysql.com/doc/>

Применение методов машинного обучения для автоматического распознавания патологий растений

А. С. Железной

Студент бакалавриата

М. А. Дрюченко

Доцент

Введение

Вопросы создания и использования информационных систем и технологий в сельскохозяйственном производстве в современных условиях становятся основополагающим фактором развития инновационной деятельности и научно-технического прогресса в отрасли. Одна из основных задач в области применения современных информационных технологий связана с созданием компьютерных систем диагностики заболеваний сельскохозяйственных культур и прогнозирования их развития. Применение автоматизированной системы диагностики позволяет использовать объективную диагностическую информацию, осуществить более раннюю и достаточно точную диагностику заболеваний растений и принять решение о проведении мероприятий по защите растений [1]. Центральное место при создании компьютерных систем диагностики заболеваний растений занимают методы и алгоритмы распознавания образов.

Данная работа посвящена анализу известных подходов к задаче диагностики фитосанитарного состояния культурных растений на основе анализа изображений их фрагментов (листьев, стеблей, плодов). Рассматриваются алгоритмы текстурного анализа, вопросы выделения устойчивых признаковых описаний изображений фрагментов растений, а также алгоритмы классификации фрагментов при диагностике патологий.

1. Обобщенная схема анализа изображений фрагментов растений с целью выявления патологий

Обобщенная схема анализа изображений, размеченных по результатам распознавания и сегментации фрагментов растений, представлена на рис. 1. Данная схема включает следующие основные этапы:

1. Анализ растрового представления картинок фрагментов заданных базовых классов объектов (листьев, стеблей, плодов) и выделение описывающих их диагностических признаков, позволяющих определить фитосанитарное состояние.

2. Определение репрезентативных диагностических признаков.

3. Формирование статистик по каждому классу объектов с использованием множества изображений здоровых и пораженных патологиями растений.

4. Настройка алгоритмов классификации: пороговых значений в решающих правилах для эвристических алгоритмов (статистических критериев) и обучение классификаторов (нейронных сетей).

5. Применение статистических критериев или обученных классификаторов для анализа фрагментов изображений тепличных культур в составе реального программно-аппаратного комплекса для визуального фенотипирования и анализа растительных патологий.

2. Формирование базовых фрагментов изображений кустов тепличных культур

Исходные данные об исследуемых растениях заданы в виде матрицы (изображений листьев). Получение распознанных представлений растительных объектов проводится на основе пороговых или обучаемых алгоритмов сегментации, однако подробное их рассмотрение выходит за рамки данной работы. Пример распознанного сегментированного представления фрагмента куста томата приведен на рис. 2. Здесь представлены объекты четырех классов - листья (светло-синие), стебли (темно-синие), плоды (желтые) и прочие фрагменты (бледно-желтые).

В рамках курсовой работы основное внимание будет уделено анализу объектов первого класса (листьев).

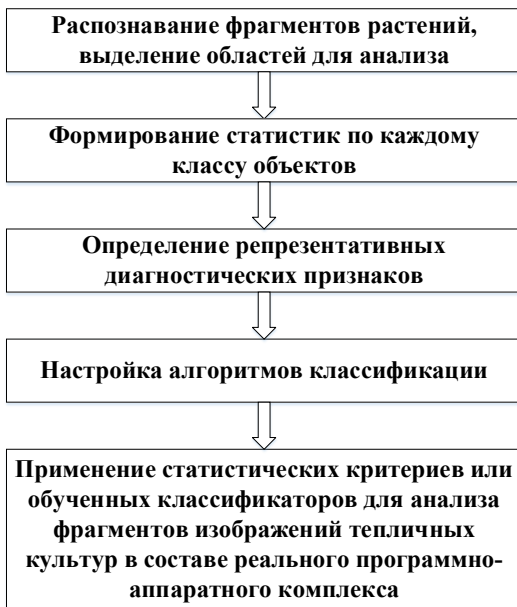


Рис. 1. Обобщенная схема



Рис. 2. Пример исходного и распознанного и сегментированного изображения фрагмента куста томата

3. Определение набора диагностических признаков по изображениям листьев

На данном этапе формируется набор диагностических признаков, которые определяются как некоторые числовые характеристики рассматриваемого фрагмента изображения, например, текстурные признаки, основанные на статистике первого порядка распределений яркостей. Следует отметить, что отдельные характеристики, вычисленные для всех фрагментов изображений, рассматриваются как отдельное пространство признаков [2].

Один из подходов к текстурной сегментации состоит в подсчете некоторой меры зернистости текстуры во всех точках изображения с последующим обнаружением изменений этой меры. Фактически исходное изображение подвергается предварительной обработке с тем, чтобы преобразовать его в яркостное изображение. Основная трудность, связанная с этим подходом, состоит в том, что параметры текстуры измеряются в некотором окне и, следовательно, их значения, полученные в окрестности границы между текстурными областями, являются усредненными. В результате трудно точно локализовать границу между текстурными областями. Другой подход к текстурной сегментации заключается в обнаружении переходной зоны между областями с различной текстурой.

В задачах распознавания изображений часто оказывается полезным разбить объект сложной формы на связанный набор частей простой, легко описываемой формы. Сегментацию формы можно выполнить в два этапа. Сначала объект произвольной формы аппроксимируется набором связанных прямолинейных или криволинейных отрезков. Затем аппроксимированная форма подвергается сегментации по точкам перегиба.

Метод синтаксического описания текстур состоит в выделении из текстуры набора связанных компонент, классифицированных по какому-либо свойству с последующим выделением групп. Далее, производя поиск соответствующих компонент на любом другом изображении, происходит детектирование уже известной текстуры. Исходя из вышесказанного следует набор требований к алгоритму сегментации:

- области должны быть однородны относительно определенных характеристик;
- смежные области должны существенно отличаться по значениям выбранных характеристик, относительно которых они считаются однородными;
- границы каждого сегмента должны быть простыми, пространственно точными.

4. Анализ изображений с использованием алгоритмов текстурного анализа

Текстура характеризуется специфическим изменением цвета в изображении или его некоторой части, а также наличием на изображении характерных линий. Анализ текстуры играет фундаментальную роль в классификации объектов и выделение значительных областей данного изображения уровня серого. Несмотря на свою вездесущность в данных изображения, текстура не имеет точного определения. Независимо от определения, большинство алгоритмов, которые выводят текстуру из изображения, попадают в категории статистической или частотной области [3].

Далее следует краткое описание текстурных методов, представляющих интерес.

5. GLCM

Пространственное распределение яркостных значений позволяет оценить статистические признаки изображения. Наиболее известным алгоритмом является Матрица яркостной зависимости (Gray-level Co-occurrence matrix).

Первый этап – вычисление матрицы яркостной зависимости, которую можно описать следующей формулой:

$$C(i, j) = \sum_{p=1}^N \sum_{q=1}^M \begin{cases} 1, & I(p, q) = i, I(p + \Delta x, q + \Delta y) = j, \\ 0 & \end{cases}$$

Матрица яркостной зависимости вычисляет, как часто пиксель с интенсивностью серого уровня i встречается в определенной пространственной связи с пикселем со значением j . Количество уровней серого на изображении определяет размер GLCM [4].

Второй этап – вычисление признаков на основе матрицы яркостной зависимости. К ним относят контраст (мера интенсивности или вариации уровня серого между опорным пикселем и его соседом), энтропию и корреляцию (эта функция показывает, как опорный пиксель связан с его соседом, 0 некоррелирован, 1 полностью коррелирован)

6. Признаки Тамуры

Тамура и др. приняли подход к разработке текстурных признаков, которые соответствуют визуальному восприятию человека. Они определили шесть текстурных признаков (зернистость, контрастность, направленность, линейность, регулярность, грубость) и сравнили их с психологическими измерениями для людей [4].

На рис. 3. представлены примеры вычисленных значений текстурных признаков для фрагментов изображений различных классов.

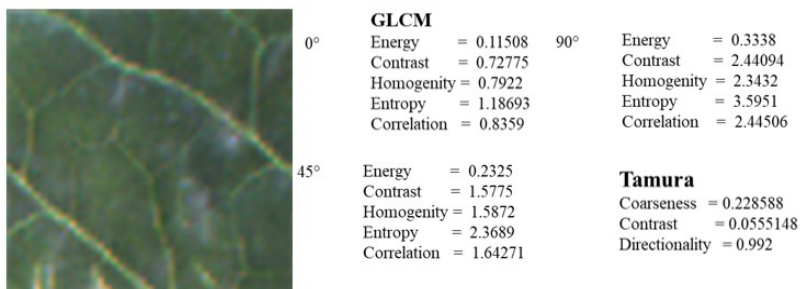


Рис. 3. Примеры вычисленных значений текстурных признаков для фрагмента изображения

7. Определение репрезентативных диагностических признаков

На этом этапе из каждого подмножества сильносвязанных признаков определяется один-единственный признак, и в результате выполнения данного этапа выделяется набор репрезентативных диагностических признаков. Основная идея выбора репрезентативных диагностических признаков заключается в их различии (несходстве) в формируемом наборе репрезентативных признаков. В процессе формирования набора репрезентативных признаков требуется, чтобы каждый выделенный признак был типичным представителем своего подмножества сильносвязанных признаков. В результате выполнения данного этапа получаем сокращенное пространство диагностических признаков, размерность которого намного меньше исходного. Для определения репрезентативных признаков могут использоваться алгоритмы случайного леса и деревьев решений.

8. Классификация фрагментов изображений с использованием нейронных сетей

В качестве классификатора при решении задачи диагностики растительных патологий, можно использовать многослойную НС, реализующую двухальтернативное решающее правило [4]. Сети данного класса отличаются: хорошими обобщающими свойствами, возможностью использования нелинейных передаточных функций для обработки линейно-неразделимых множеств значений признаков, возможностью дообучения НС-классификаторов при поступлении новых данных или обучения новых НС для обработки специфических данных, простотой реализации и практического использования.

При рассмотрении полного перечня классов анализируемых объектов (листьев, стеблей, плодов) целесообразно использование трех

НС, каждая из которых обучается для классификации объектов своего типа. Архитектура НС-классификатора представлена на рис. 4.

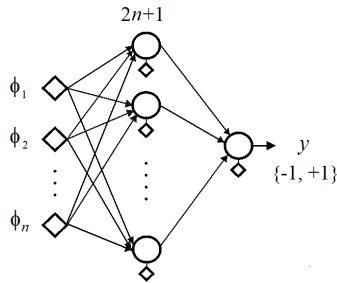


Рис. 4. Архитектура нейронной сети, используемой для диагностики растительных патологий

9. Описание разработанного программного модуля, реализующего распознавание патологий растений и его тестирование

Программный модуль был написан на языке Python версии 3.7. В начале работы использовался Jupyter Notebook, затем был сделан выбор в сторону платформы Google Colab, предоставляющей возможность использовать 12 Гб видео памяти на стороне Google, а также использовать все необходимые библиотеки и фреймворки без дополнительной установки. Благодаря работе на Google Colab удалось сократить обучение модели, в отдельном случае, с 6 часов до 30 минут.

В качестве исходных данных использовался большой датасет изображений листьев растений томатов. В общем случае этот набор включал около 10000 изображений и 9 вариантов различных патологий.

При создании модели по умолчанию использовалась архитектура «channel_last», но также был создан переключатель для бэкэндов, которые поддерживают «channel_first». Был добавлен первый CONV => RELU => POOL. Слой CONV имеет 32 фильтра с ядром 3 x 3 и активацией RELU (Rectified Linear Unit). Отсев составляет 25%. Затем были созданы два набора блоков (CONV => RELU) * 2 => POOL. После - только один набор слоев FC (Fully Connected Layer) => RELU.

Для тестирования разработанного программного модуля было проведено несколько экспериментов с различными исходными данными. В процессе этих экспериментов менялось:

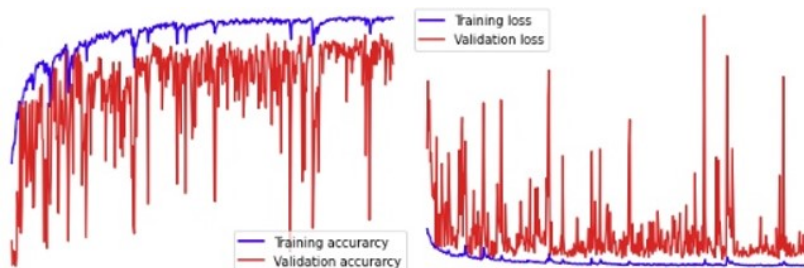
- количество эпох.
- количество исходных изображений.
- количество шагов в одной эпохе.

На рис. 5. представлены примеры исходных изображений.



Рис. 5. Примеры сегментированных изображений из датасета

На рис. 6. можно увидеть самый точный результат тестирования разработанного модуля.



Полученная точность тестирования – 96.375

Рис. 6. Результат тестирования разработанного программного модуля.

Заключение

В результате выполнения данной работы была изучена предметная область обработки цифровых изображений, рассмотрены известные алгоритмы выделения характеристических признаков фрагментов изображений, текстурного анализа, алгоритмы определения репрезентативных диагностических признаков и классификации фрагментов изображений на «хорошие» и «подверженные патологиям». Также был разработан тестовый вариант программного модуля.

В дальнейшем планируется использование данного модуля в системе распознавания фрагментов изображений кустов томатов с патологиями.

Список литературы

1. Лбов, Г. С. Устойчивость решающих функций в задачах распознавания образов и анализа разнотипной информации / Г. С. Лбов, В. Б. Бериков. – Новосибирск : ИМ СО РАН, 2005. – 218 с.
2. Журавлев, Ю. И. Избранные научные труды / Ю. И. Журавлев. – М. : Магистр, 1998. – 420 с.
3. Журавлев, Ю. И. Распознавание. Математические методы. Программная система. Практические применения / Ю. И. Журавлев, В. В. Рязанов, О. В. Сенько. – М. : Фазис, 2006. – 159 с.
4. Вудс, Р. Цифровая обработка изображений / Р. Вудс, Р. Гонсалес. – М. : Техносфера, 2005. – 752 с.

Профилактика управления продажами коммерческой фирмы

С. В. Захаров

Студент бакалавриата

И. В. Абрамов

Доцент

Введение

В условиях информатизации экономики и ускорения бизнес-процессов, а также возрастания конкуренции, от компаний, чья деятельность направлена на получение прибыли, требуется гибкость и инновационность в сфере продаж. Умение и возможность среагировать на изменения в запросах потребителей и диспозиции конкурентов – незаменимые качества для любого отдела продаж.

С развитием деловых отношений, роль продавца преобразилась из обработчика заказов в более многокритериальную роль, подразумевающую налаживание и ведение партнёрских отношений с клиентом, а также создание базы знаний, грамотное структурирование и оперирование которой позволит увеличить эффективность продаж.

Управление продажами – менеджмент обеспечения выполнения маркетинговых задач в сфере продаж товаров и услуг. Понятие управления продажами в настоящий момент трактуется в разных вариациях, прежде всего, по двум причинам: усложнение задач, стоящих перед менеджером в этой области, а также наличие различных точек

зрения на это понятие (экономическая, социальная, психологическая, маркетинговая и другие).

Универсальность применения разработок подчёркивает актуальность исследований в данной области: проблема повышения эффективности продаж стоит перед руководством и менеджментом любой компании, вне зависимости от масштаба бизнеса и сферы деятельности.

В мировой практике формирование методов и процессов управления продажами прошло через несколько этапов, на каждом из которых учитывались постулаты предыдущих периодов, а также новые возникающие требования. Постепенно, компании пришли к тому, что обсуждение тактик и стратегий продаж стало происходить на высоких уровнях управления, что свидетельствует о приоритетности этой подструктуры менеджмента над другими.

Задача менеджера по продажам – создание среды, в которой деятельность по повышению эффективности продаж будет максимально продуктивной.

Особое место в улучшении процесса продаж занимает процедура автоматизации отдельных этапов или деятельности компании в целом. Развитие информационных технологий позволяет находить новые подходы к решению управленческих задач.

Целью данной работы будет анализ деятельности коммерческой организации с целью определения путей повышения эффективности продаж.

1. Анализ факторов предприятия

Для рассмотрения процесса продаж будет уместно рассмотреть текущую ситуацию в предприятия в целом. Кроме того, необходимо учесть влияние факторов внешней среды [1].

Для этого будем использовать SWOT-анализ – метод стратегического планирования, который применяется для определения внешних и внутренних факторов среды предприятия и дифференцирования их по категориям:

– Strengths (сильные стороны предприятия) – уникальные возможности и ресурсы, а также аспекты, которые являются преимуществами с точки зрения конкурентов;

– Weaknesses (слабые стороны предприятия) препятствуют развитию компании или её оптимальному функционированию. Это те параметры, по которым фирма уступает другим фирмам либо испытывает нехватку ресурсов. Также сюда можно отнести те аспекты, которые воспринимаются конкурентами как слабости;

– Opportunities (внешние возможности) представляют собой факторы, которые, вероятно, могут быть использованы как дополнительные возможности для достижения цели. К ним относятся текущие тренды или перспективы, которые организация способна конвертировать в преимущества;

– Threats (внешние угрозы) могут затруднить достижение поставленных задач.

Для построения таблицы сильных и слабых сторон был проведён опрос с участием тех сотрудников предприятия, которые в той или иной степени связаны с процессами продаж, рекламы и контроля. Для этого были выбраны следующие должности: генеральный директор, советник генерального директора, специалист по работе с клиентами, специалист по продажам и дизайнер-рекламщик. Перед этим, на основе анализа мнений всех работников о деятельности компании, а также с учётом общих практик бизнес-организации, были составлены списки для каждой из четырёх групп факторов [2].

Сотрудникам было предложено оценить важность каждого фактора по пятибалльной шкале. Результаты были зафиксированы в таблице, изображённой на рисунке. Далее, для каждого фактора было посчитано медианное значение оценки по формуле:

$$M_e = x_{M_e} + i_{M_e} \frac{\frac{\sum f}{2} - S_{M_e-1}}{f_{M_e}}, \quad (1)$$

где x_{M_e} – нижняя граница медианного интервала, i_{M_e} – ширина медианного интервала, $\frac{\sum f}{2}$ – количество всех значений, делённое на 2, S_{M_e-1} – суммарное количество наблюдений, которое было накоплено до начала медианного интервала, т.е. накопленная частота предмедианного интервала, f_{M_e} – число наблюдений в медианном интервале [3].

На основе вычислений в табличном процессоре Microsoft Excel получим таблицу ранжировок факторов. Пример расчёта и ранжировки представлен на рисунке.

Для дальнейшего исследования были выбраны 3-5 факторов, получивших самые высокие медианные оценки.

2. Создание матрицы решений

Согласно методике, при анализе полученных данных и выработке решений есть четыре варианта возможных стратегий:

- S+O (стратегия прорыва) – сопоставление преимуществ и возможностей, как сильная сторона компании может быть применима к перспективе;
- S+T – использование сильных сторон для устранения или профилактики угроз;
- W+O – соотнесение слабых сторон и возможностей: устранение слабостей бизнеса, преобразовав их в сильные стороны;
- W+T (стратегия выживания) – нахождение взаимосвязи между устранением слабостей и возможным последующим эффектом снижения рисков от угроз [4].

№	S - Сильные стороны	Оценки экспертов					медиана
		Э1	Э2	Э3	Э4	Э5	
3	Гибкость команды	5	4	5	4	5	5
4	Мобильность предприятия	5	4	5	5	5	5
5	Хорошее понимание потребителей	4	5	5	5	5	5
6	Удобное месторасположение	5	5	3	5	5	5
8	Надёжное, работающее на перспективу	5	4	5	4	5	5
10	Быстрая реакция на требования заказчика	5	5	5	4	5	5
2	Широкий асс-т услуг	4	3	4	5	5	4
7	Низкие цены	4	3	5	4	4	4
9	Массовость целевой аудитории	4	5	2	3	4	4
1	Оформленность бренда	3	5	4	3	2	3

Рисунок. Ранжировка факторов (сильные стороны)

Составим матрицу решений A_{ij} , где i – строки с выделенными сильными/слабыми сторонами внутренней среды организации, j – столбцы с выделенными возможностями/угрозами внешней среды, a_{ij} – перекрестие двух факторов.

Перекрестие может быть пустым (в случае, если факторы не взаимосвязаны или взаимоисключающие), либо может содержать одну из следующих категорий решений: элемент стратегии, рекомендация, предпосылка к разработке или описание конкретной идеи для разработки решения по данному пересечению.

В результате составления перекрёстной матрицы были выработаны следующие возможные решения для компании:

- проведение рекламной кампании;
- создание корпоративной среды для взаимодействия с предпринимателями – потенциальными клиентами;
- разработка системы управления продажами;
- создание базы знаний;

- комплекс мер по налаживанию благоприятных постоянных отношений с клиентами, а также профилактика ухода постоянных клиентов к другому заказчику;
- и другие.

Заключение

В рамках данной работы был рассмотрен вопрос повышения эффективности продаж в коммерческой фирме посредством анализа внутренней и внешней среды организации методом SWOT с выработкой решений или идей для разработок, которые будут призваны оптимизировать процесс продаж или другие процессы, непосредственно связанные с продажами.

Был проведён опрос сотрудников компании, на основе которого была составлена матрица решений SWOT, состоящая из наиболее весомых, на взгляд сотрудников, факторов среды, и решений по каждому пересечению сильных/слабых сторон и возможностей угроз.

В качестве перспективного направления работы можно рассмотреть внедрение математических методов в SWOT-анализ, а также реализация решений, выработанных в процессе анализа.

Список литературы

1. Зозулёв, А. В. Маркетинговые исследования : теория, методология, статистика : Учеб. пособие / А. В. Зозулёв, С. А. Солнцев. – М. : Рыбари, 2008. – 644 с.
2. Савицкая, Г. В. Анализ хозяйственной деятельности предприятия : Учеб. пособие / Г. В. Савицкая. – 7-е изд., испр. – Мн. : Новое знание, 2002. – 704 с.
3. Гмурман, В. Е. Теория вероятностей и математическая статистика: Учеб. Пособие для вузов / В. Е. Гмурман. – 9-е изд., стер. – М. : Высш. шк., 2003. – 188 с.
4. Токарев, В. А. Применение SWOT-анализа при разработке стратегии фирмы / Токарев В. А. – М. : Русский менеджмент, 2004. – 60 с.

Алгоритм распределения канальных ресурсов в сетях когнитивного радио

М. И. Игнатов

Студент бакалавриата

А. А. Головкин

Ассистент

Введение

Развитие радиовещания, цифрового телевидения и беспроводных систем связи происходит столь стремительно, что все более остро встает вопрос об эффективности использования частотного спектра.

В настоящее время в связи с бурным развитием технологий беспроводной связи, а также существующей потребностью организации устойчивой мультисервисной сети связи из большого количества радиостанций, имеется ряд актуальных задач: во-первых, возникает проблема генерации настроечных данных для радиостанций таким образом, чтоб обеспечить электромагнитную совместимость радиостанций, во-вторых – ввода этих настроечных данных в радиостанции за приемлемое время и с минимальными усилиями со стороны пользователей и администраторов сети связи, в-третьих – существует проблема с тем, чтобы при выходе из строя одного из узлов системы остальные узлы продолжали функционировать и в-четвертых использования ресурса пропускной способности сети для совместной передачи трафика различного вида.

Распределение спектра основывается на выделении конкретного диапазона частот для конкретной услуги. Однако, большая часть выделенного диапазона радиочастот используется от случая к случаю, что приводит к неэффективному использованию частотного ресурса. Проблема неэффективного использования спектра может быть решена за счет новой системы доступа к лицензированным полосам частот в которых работают существующие (так называемые первичные) пользователи. Такой новой системой доступа является динамический доступ к спектру [1]. Основной технологией, использующей динамический доступ к спектру, является когнитивное радио, которое дает возможность получить доступ к беспроводному каналу наравне с первичными пользователями. Термин "когнитивное радио" был впервые введен Митолой и Магуайром в [2]. В данной статье когнитивное радио

определяется как радио, которое воспринимает окружение, в котором оно находится, и в результате может адаптировать параметры связи в соответствии с этим пониманием. В настоящее время, термин «когнитивное радио» приобрел более широкое значение.

1. Модель сети

Для того, чтобы получить доступ к радиочастотному спектру, в котором работают лицензированные (первичные) пользователи, не нарушая их прав, с требуемым качеством обслуживания, каждый пользователь когнитивного радиоприбора в сетях когнитивного радио должен:

- определить доступную часть спектра;
- выбрать лучший из доступных каналов;
- скоординировать доступ к этому каналу с другими пользователями;
- освободить канал, когда возобновит работу лицензированный пользователь.

Одной из основных задач реализации такого подхода является выбор канала приёма/передачи.

Рассмотрим представленную на рис. 1 модель беспроводной сети, представляющую собой совокупность приемников и передатчиков, образующих пары.

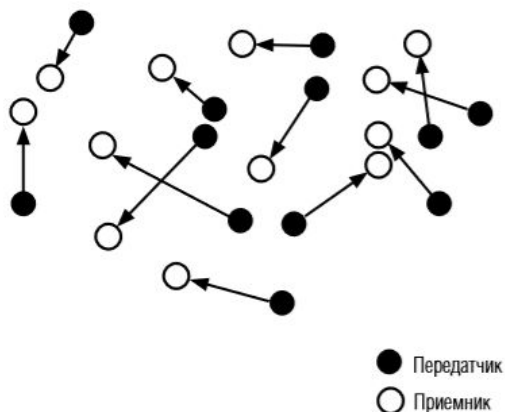


Рис. 1. Беспроводная модель приемепередающих пар

В данной модели все приемепередающие пары находятся в одном кластере, т.е. передача информации от передатчика к приемнику осуществляется напрямую без использования промежуточных узлов. В

сети может использоваться определенное количество каналов связи для передачи информации от передатчика к приемнику. При этом каждая приемопередающая пара может использовать не более одного канала связи.

Нужно адаптировать приемопередающие пары таким образом, чтобы каналы связи для конкретных пар были оптимальными. Для этого используем предложенную в [3] целевую функцию:

$$U_i(a_i, a_{-i}) = \sum_{k=1, k \neq i}^n p_k g_{ki} f(a_k, a_i) + \sum_{k=1, k \neq i}^n p_i g_{ik} f(a_i, a_k) \quad (1)$$

где a_i – канал связи, выбранной i -той парой, a_{-i} – каналы связи, выбранные остальными парами, p – мощность передатчика, g – коэффициент передачи радиочастотного тракта между каким-либо передатчиком и приёмником, f – функция, принимающая значение, равное 1, если её аргументы – выбранные каналы связи двух различных пар приёмников и передатчиков – совпадают, и равное 0, если не совпадают. Таким образом, целевая функция (1) для i -той пары представляет собой сумму помех, наводимых остальными передатчиками на i -м приёмнике, и помех, наводимых i -м передатчиком на остальных приёмниках. Критерием выбора канала связи для данной целевой функции является её минимум.

2. Алгоритм и результаты моделирования

Для имитационного моделирования распределения канальных ресурсов в сети приемопередающих пар на основе модели с целевой функцией (1) была разработан и реализован алгоритм (рис. 2) на языке Python.

Алгоритм заканчивает работу и распределение каналов приемопередающих пар считается окончательным тогда, когда в очередном раунде ни одна из пар не поменяет выбранный в предыдущем раунде канал связи.

Моделирование проводилось при следующих значениях параметров: количество приемопередающих пар $n=5-50$; количество каналов связи $m=3-10$; приемопередающие пары распределены случайным образом на плоскости размером 400×400 м²; мощности всех передатчиков равны 1 Вт; исходное распределение каналов связи между приемопередающими парами также случайно.

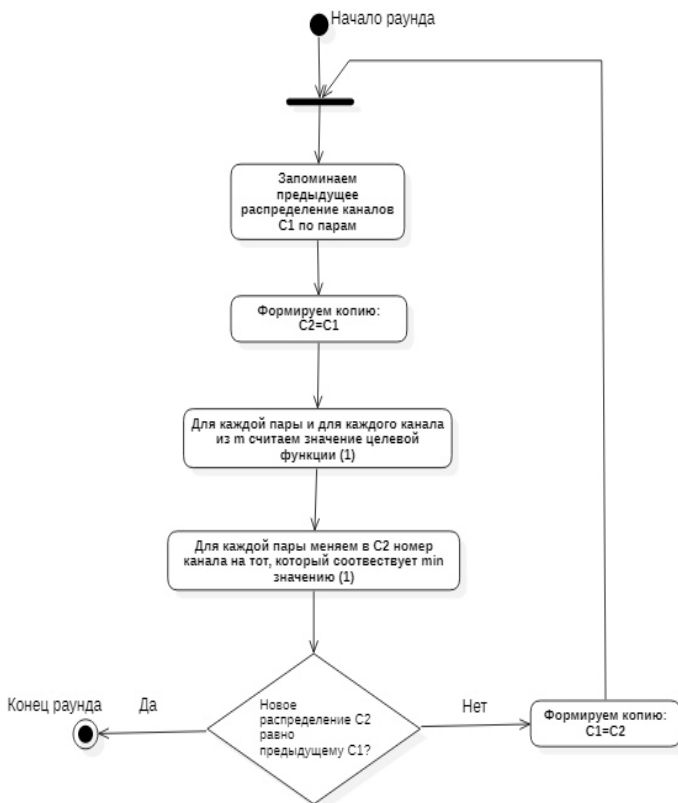


Рис. 2. Диаграмма состояний алгоритма распределения каналов

На рис. 3 представлены распределения каналов связи среди приемопередающих пар до и после работы алгоритма для случая $n=10$, $m=4$. Оси отражают расположение приемопередающих пар на плоскости, первичный и адаптированный каналы связи указаны в скобках у каждой пары.

Для качественной оценки адаптации каналов приёмопередающих пар в соответствии с предложенным алгоритмом использовалось отношение сигнал/шум для приёмника каждой пары (в дБ). Исходное распределение каналов для 10 пар и соответствующие отношения сигнал/шум (левые столбцы), а также отношения сигнал/шум после адаптации каналов (правые столбцы) представлены на рис. 4. Можно отметить, что при заданных условиях моделирования отношение

сигнал/шум практически для каждой пары выросло в среднем на 3-5 дБ, а работа алгоритма заканчивается через 3 раунда.

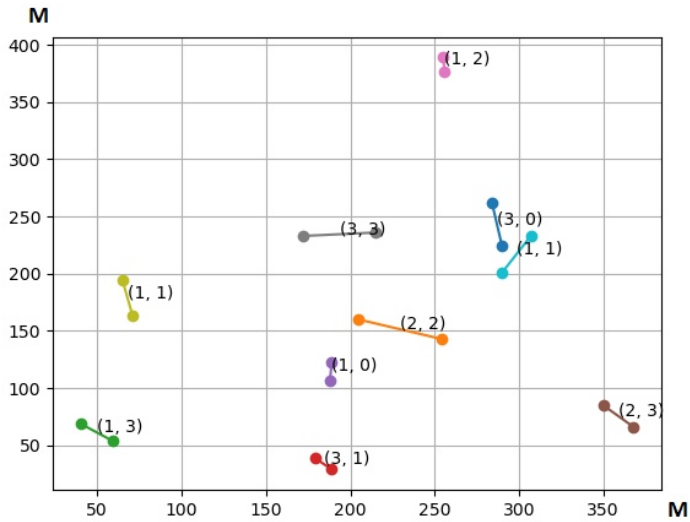


Рис. 3. Распределение каналов связи приемопередающих пар на плоскости

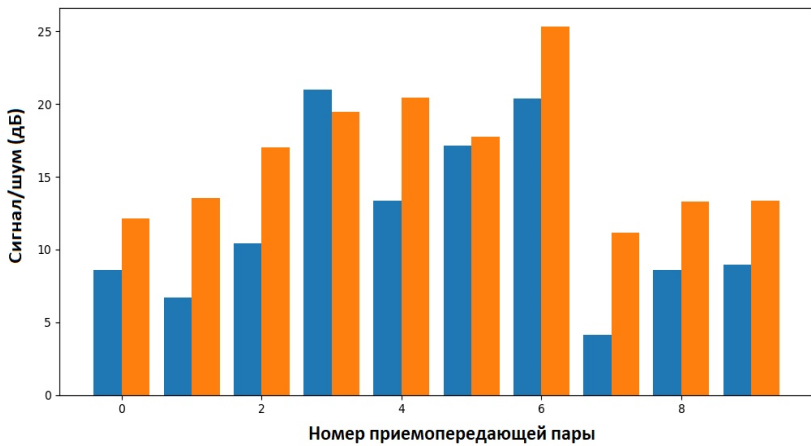


Рис. 4. Отношение сигнал/шум до и после работы алгоритма

Заключение

В данной статье рассмотрен алгоритм распределения каналов связи в сетях приемопередающих пар, в основе которого лежит минимизация уровня помех на приёмниках. Так же приведены результаты моделирования распределения каналов связи на основе рассмотренного алгоритма. Реализация алгоритма и последующее моделирование позволили выявить следующие аспекты его функционирования: алгоритм чувствителен к первичному распределению каналов, что отражается в виде увеличения/уменьшения отношения сигнал/шум после адаптации каналов связи; при больших значениях n (более 15) и m (более 5-8) для некоторых приёмопередающих пар наблюдается значительное ухудшение отношения сигнал/шум (15-30 дБ); для n не более 30 алгоритм заканчивает работу в среднем за 3-6 раундов.

Рассмотренный алгоритм может быть использован в совокупности с другими методами оценки и выбора каналов связи, в частности, в перспективных направлениях построения сложных алгоритмов обработки информации с помощью аппарата искусственных нейронных сетей (НС) [4, 5].

Список литературы

1. Doyle, L. Essentials of Cognitive Radio / L. Doyle. – Cambridge : Cambridge University Press, 2009. – 252 с.
2. Mitola, Joseph. Cognitive Radio. An Integrated Agent Architecture for Software Defined Radio / Joseph Mitola. Dissertation ... of Doctor of Technology. Sweden, Stockholm : Royal Institute of Technology (KTH), 2000. – 313 с.
3. Ошмарин, Д. В. Распределение канальных ресурсов в сетях когнитивного радио на основе теории игр / Д. В. Ошмарин // Бизнес-информатика. – 2010. – №4 (14). – С. 38-45.
4. Кирсанов, Э. А. Обработка информации в пространственно-распределенных системах радиомониторинга: статистический и нейросетевой подходы / Э. А. Кирсанов, А. А. Сирота. – М. : Физматлит, 2012. – 344 с.
5. Фомин, Л. А. Нейрофизиологическая модель управления распределенной системой / Л. А. Фомин, П. А. Будко, Д. В. Шлаев // Нейрокомпьютеры : разработка, применение. – 2008. – №7. – С. 62-68.

Моделирование распространения инфекционного заболевания COVID-19 на основе данных по Воронежской области с помощью SIR-модели

А. И. Кищенко

Студент бакалавриата

В. К. Каверина

Доцент

Введение

В 2020 году мир столкнулся с проблемой. Это пандемия COVID-19. Первыми заболевшими были жители города Ухань, Китай. Далее появились инфицированные в Европе и Америке. В России первый зарегистрированный случай был в начале марта, в Воронеже – 19 марта. Именно поэтому темой было выбрано исследование изменения количества заболевших по Воронежской области.

1. История модели SIR

Для начала рассмотрим модель, с помощью которой проводились расчеты. Как указано в заголовке статьи, название этой модели SIR. Она была описана в 1927 году У. О. Кермаком и А. Г. Маккендриком. Модель применяли для анализа распространения различных заболеваний, особенно кори, свинки, краснухи – детских болезней, в ходе которых вырабатывается пожизненный иммунитет. SIR – это аббревиатура от «Susceptible-Infected-Recover». Восприимчивые (Susceptible) – это группа людей, которые уязвимы к инфекционным болезням и могут в любой момент перейти в группу зараженных. Зараженные (Infected) – это группа людей, которые болеют и способны заразить восприимчивую группу, но в конце концов они перейдут в группу восстановленных. Восстановленные (Recover) – это группа людей, которые переболели и приобрели иммунитет. Эта модель развивалась, и у нее появились модификации. В качестве примера можно назвать модель SIRS – «Susceptible-Infected-Recover-Susceptible». Её отличие в том, что через какое-то время иммунитет ослабевает и человек снова переходит в группу восприимчивых. Или SEIR, где E – Контактные (Exposed). Перед стадией зараженных выделяется стадия контактных, у которых болезнь проходит инкубационный период.

2. Исследование модели SIR

Модель SIR [1] представляет собой нелинейную автономную систему дифференциальных уравнений следующего вида:

$$\begin{cases} \frac{dS}{dt} = -\beta IS \\ \frac{dI}{dt} = \beta IS - \gamma I \\ \frac{dR}{dt} = \gamma I, \end{cases} \quad (1)$$

где $S(t)$ – количество восприимчивых к заражению людей в момент времени t , $I(t)$ – количество зараженных (и способных заражать других) в момент времени t , $R(t)$ – количество переболевших, обладающих иммунитетом людей в момент времени t . Здесь введены следующие ограничения: постоянство объема популяции, иными словами, болезнь не считается смертельной и ее период настолько мал по сравнению с продолжительностью жизни человека, что не учитывается смертность и рождаемость; промежуток времени, когда человек остается опасным для окружающих, распределен по экспоненциальному закону, контакты случайные и равновозможные. Третье уравнение данной системы избыточно, так является следствием из первых двух уравнений при выполнении условия постоянства объема популяции.

Исследуем модель, состоящую из двух первых уравнений системы (1). Найдем особые точки. Для этого необходимо приравнять правые части системы к нулю. В результате получим точки

$$A(0,0), B\left(\frac{\gamma}{\beta}, 0\right).$$

Для того чтобы определить тип особых точек, линеаризуем систему (1) и составим матрицу частных производных D :

$$D = \begin{pmatrix} -\beta I & -\beta S \\ \beta I & \beta S - \gamma \end{pmatrix}.$$

Корни характеристического уравнения, составленного для точки A , имеют вид:

$$\lambda_1 = 0, \quad \lambda_2 = -\gamma < 0.$$

Мы видим, что точка A вырожденная.

Для точки B

$$\lambda_1 = i\sqrt{\gamma}, \quad \lambda_2 = -i\sqrt{\gamma}.$$

Точка B имеет тип «центр», так как корни характеристического уравнения чисто мнимые.

Согласно теореме Пуанкаре [2], особая точка B нелинейной системы (1) будет иметь тип «фокус» или «центр». Построим фазовый портрет системы (1). Для этого разделим второе уравнение на первое уравнение.

$$\frac{dI}{dS} = -1 + \frac{\gamma}{\beta} \frac{1}{S}. \quad (2)$$

Общий интеграл дифференциального уравнения (2) имеет вид:

$$I(S) = -S + \frac{\gamma}{\beta} \ln|S| + \text{const}. \quad (3)$$

Фазовые траектории изображены на рис. 1. Очевидно, что они незамкнуты, так как $\lim_{S \rightarrow +\infty} I = -\infty$, а значит, особая точка B нелинейной системы (1) будет иметь тип «фокус».

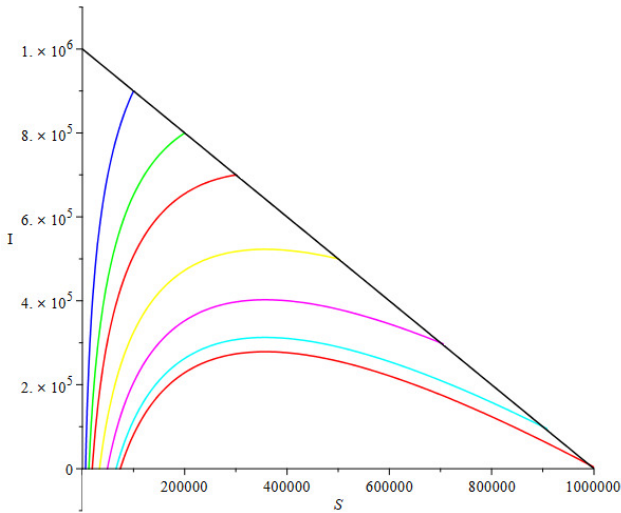


Рис. 1. Фазовый портрет системы (1)

3. Построение модели SIR

Построим модель и оценим ее параметры для Воронежской области. Как известно, построение модели начинается со сбора и анализа данных, поэтому были взяты данные по количеству заболевших

в период с 19 марта по 4 мая по Воронежской области. Данные представлены на рис. 2.

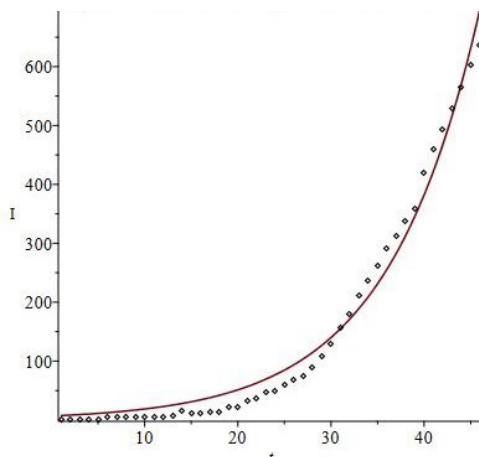


Рис. 2. График числа заразившихся по Воронежской области: точками показаны реальные данные, кривой – смоделированные

В начале эпидемии количество инфицированных изменяется по экспоненциальному закону. С помощью пакета Wolfram Mathematica была построена функция $I(t)$, зависимость числа инфицированных от времени, которая также изображена на рис. 2:

$$I(t) = 6.99869 e^{0.100425(t-0.161899)}. \quad (4)$$

Одним из важных показателей модели SIR является основной коэффициент воспроизводства R_0 :

$$R_0 = \frac{dS}{dR} = \frac{\beta}{\gamma}. \quad (5)$$

Его еще можно понимать как среднее число людей, которых успевает заразить один инфицированный. Если $R_0 < 1$, то эпидемия затухает, если $R_0 > 1$, то эпидемия разрастается. Например, при $R_0 = 2$ количество заболевших примерно 80%. Для нахождения этого коэффициента воспользуемся более точной формулой:

$$R_0 = \left(1 + \frac{\tau_1}{\tau_2} \ln 2\right) \left(1 + \frac{\tau_i}{\tau_2} \ln 2\right), \quad (6)$$

где $\tau_1 = 5$ – инкубационный период болезни, $\tau_2 = 3$ – период, когда человек заразен, но у него не наблюдается симптомов, $\tau_3 = 7$ – период, за который количество заболевших удваивается. Последний параметр был найден с помощью функции (4). В итоге получили $R_0 = 1.9$, значит, количество заболевших приблизительно будет 80%.

Для построения модели SIR найдем параметры β, γ :

$$\beta = \frac{1}{T}, \quad \gamma = cp, \quad (7)$$

где $T = 14$ – время, за которое человек выздоравливает, т.е. продолжительность болезни, $c = K / N$ – коэффициент отношения числа контактов (K) к общей численности популяции (N), p – вероятность заражения.

4. Анализ решения модели SIR

На языке C# была создана программа построения решения системы (1). Изменением параметров K и p – тех, которые человек может контролировать, – были получены графики, представленные на рис. 3, 4.

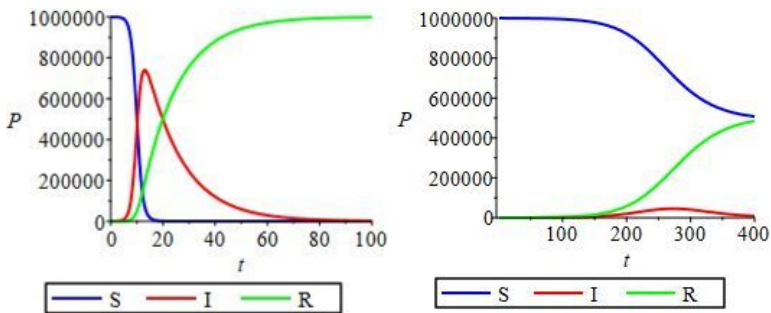


Рис. 3. Графики решения SIR: слева количество контактов одного человека в день равно 100, справа – 10

На рис. 3 постоянной оставалась вероятность заражения, изменялось только количество контактов в день. На левом графике можно увидеть резкое увеличение количества заболевших – около 75%, примерно в течение 3 месяцев эпидемия затухает, так как больше не осталось восприимчивых. На правом графике эпидемия протекает гораздо медленнее, больше года, количество заболевших не превышает 5%. Эти графики показывают разницу в характере протекания эпидемии при нарушении и соблюдении режима самоизоляции, соответственно.

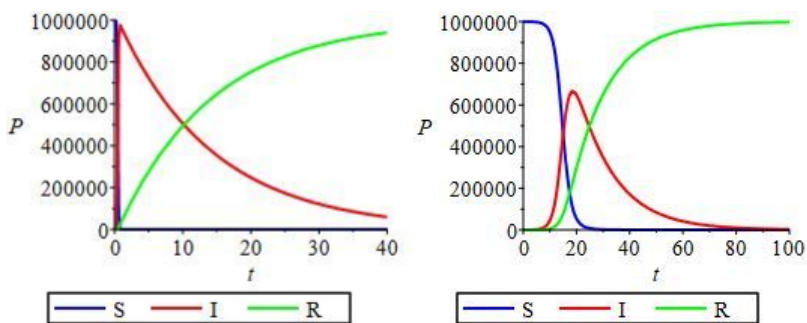


Рис. 4. Графики решения SIR: слева вероятность заражения 0.2, справа – 0.007

На рис. 4 постоянным оставалось количество контактов в день, изменялась только вероятность заражения. Как видно на графиках, при высокой вероятности заболеть количество инфицированных достигает практически 100% буквально за неделю, при низкой вероятности максимум будет достигнут лишь на двадцать первый день. Вероятность заражения зависит от нескольких факторов. Один из них – соблюдение личной гигиены. Графики наглядно показывают, как от выполнения этого простого условия меняется эпидемиологическая картина.

Заключение

Данная статья посвящена исследованию поведения модели SIR на основе данных COVID-19 по Воронежской области. Модель SIR исследована на особые точки, найдены параметры модели. В итоге разработки получена программа для построения решения SIR-модели.

Список литературы

1. Трубецков, Д. И. Феномен математической модели Лотки-Вольтерры и сходных с ней / Д. И. Трубецков // Известия вузов «ПНД». – 2011. – Т. 19 (№ 2). – С. 69-88.
2. Боровских, А. В. Лекции по обыкновенным дифференциальным уравнениям: учебник и практикум для академического бакалавриата / А. В. Боровских, А. И. Перов. – М. ; Ижевск : НИЦ «Регулярная и хаотическая динамика» ; «Институт компьютерных исследований», 2004. – 540 с.

Применение методов поиска геодезического расстояния для задачи понижения размерности 3D-модели с произвольной топологией

И. М. Косенко

Студент бакалавриата

В. С. Тарасов

Ассистент

Введение

Современные вычислительные возможности позволяют визуализировать 3D-модели с большим количеством вершин и полигонов. Однако часто возникают задачи, при решении которых необходимо упростить трехмерный объект, передав максимально точно его форму. Например, в компьютерных играх при отрисовке моделей, находящихся далеко от игрока, иногда понижают размерность объекта, т. е. уменьшают количество полигонов, его описывающих.

Некоторые алгоритмы [1], решающие данную задачу, основаны на повторяющемся объединении пары вершин в одну. Причем точка, полученная в результате слияния, всякий раз имеет новое положение в пространстве. Таким образом, топология итоговой модели сильно отличается от оригинальной. На рис. 1 схематично показано, как происходит объединение пары вершин в одну при понижении размерности модели.

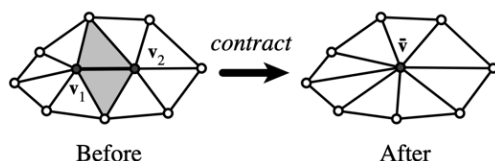


Рис. 1. Объединение двух вершин в одну при понижении размерности модели

Метод же, описываемый в данной статье, решает задачу понижения размерности путем одного лишь удаления вершин. Топология итоговой модели получается ближе к исходным данным, потому что все оставшиеся вершины сохраняют свое положение в пространстве.

Такой метод может понадобиться при решении задач, в которых важную роль играет топология модели. Например, при построении графа деформации для задачи неригидной регистрации.

2. Описание алгоритма

Перед описанием алгоритма обратим внимание на то, что в ряде случаев может понадобиться провести триангуляцию полигональной сетки. Это может быть необходимо для работы некоторых используемых алгоритмов.

Метод основан на последовательном обходе полигональной сетки. На первом шаге задается A – множество доступных для рассмотрения вершин и заполняется всеми вершинами модели. На каждой итерации цикла из текущего узла полигональной сетки каким-то образом оценивается, насколько далеко находятся другие вершины. Те из них, расстояние до которых меньше оговоренной ранее константы, т.е. те, что оказались внутри окружности определенного радиуса, удаляются из A . В конце итерации из A выбирается новая вершина, и для нее повторяются все действия. Цикл должен продолжаться до тех пор, пока A не станет пустым.

Ключевой момент заключается в том, как именно оценивать расстояние между вершинами. Об этом и пойдет далее речь.

3. Применение алгоритма Дейкстры

Заметим, что с математической точки зрения полигональную сетку можно представить в виде неориентированного графа. Это означает, что для оценки расстояния между ее узлами можно использовать алгоритм Дейкстры.

Его применение дает множество преимуществ. Так, например, алгоритм Дейкстры обладает замечательным свойством: в момент обращения к ближайшей вершине расстояние до нее, оцененное заранее, является искомым. На практике это означает, что можно прекращать обход в графа в тот момент, когда ближайшая еще не посещенная вершина лежит вне окружности оговоренного радиуса.

Однако использование алгоритма Дейкстры дает и ряд минусов. Так, метод основан на обходе графа, и, как показано на рис. 2, результат сильно зависит от топологии модели.

Справиться с возникшей проблемой можно с помощью вычисления геодезического расстояния между вершинами. Для решения этой задачи существует большое количество методов [2]. Однако самыми распространенными на данный момент являются Fast Marching и Heat.

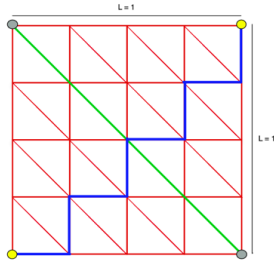


Рис. 2. Демонстрация зависимости алгоритма Дейкстры от топологии модели

4. Применение алгоритмов поиска геодезического расстояния

Fast Marching [3] – довольно известный алгоритм, поэтому можно опустить детали его реализации. Следует отметить только, что метод крайне похож на алгоритм Дейкстры и также основан на обходе графа. Основное отличие заключается в том, как происходит вычисление расстояния до вершин.

Результат решения задачи понижения размерности 3D-модели с помощью Fast Marching можно увидеть на рис. 3. Точками на модели обозначены вершины, оставшиеся после работы метода.

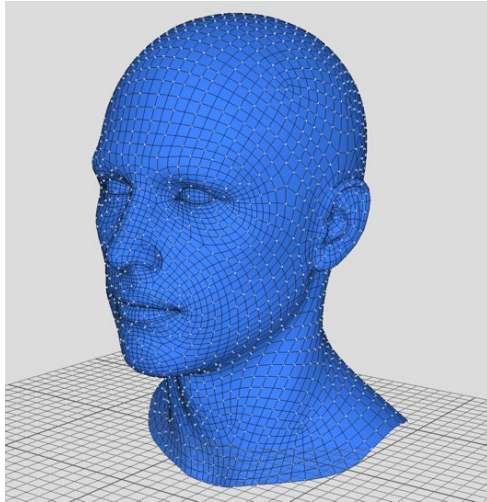


Рис. 3. Результат работы алгоритма понижения размерности модели с использованием Fast Marching

Другой метод, Heat [4], связывает функцию расстояния с функцией распространения тепла. Автор предлагает поместить в точку, из которой ищется геодезическое расстояние, неиссякаемый горячий источник. Решая уравнение теплопроводности для поверхности 3D-модели и переходя к функциям, чьи градиенты параллельны геодезическим линиям, можно найти искомые расстояния.

На рис. 4 изображено распространение тепла вдоль поверхности примитивных 3D-моделей. Точками на объектах обозначены положения источников.

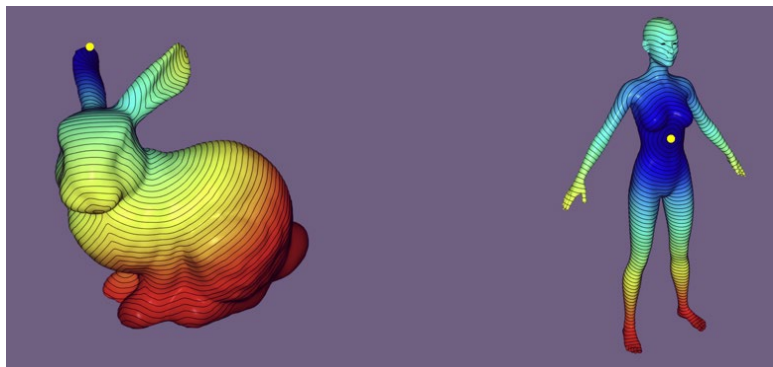


Рис. 4. Результат работы Heat на примитивных моделях.

5. Работа алгоритма на моделях с неманифолдной топологией

Следует заметить, что с добавлением Heat при попытке применить алгоритм к моделям с неманифолдной топологией могут возникнуть трудности. Основной цикл метода никак не зависит от типа полигональной сетки. Алгоритмы Дейкстры и Fast Marching также могут работать с неманифолдной топологией, ведь они оба основаны на обходе графа. Однако для работы Heat необходимо реализовать на рассматриваемом множестве градиент, дивергенцию и лапласиан. Поэтому при реализации оригинальную версию метода пришлось переписать так, чтобы дискретизация была проведена с учетом произвольности топологии.

На рис. 5 показан результат работы нашего алгоритма на примитивной модели с неманифолдной топологией.

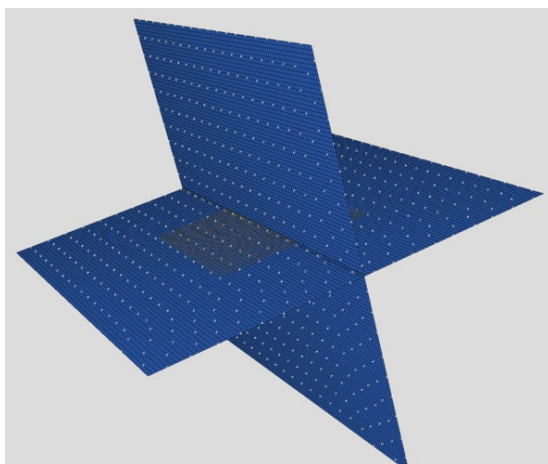


Рис. 5. Результат работы метода на модели с неманифолдной топологией

Заключение

Для подведения итогов были проведены тесты, целью которых было узнать, с применением какого из алгоритмов поиска расстояния до вершин удастся достичь наилучших результатов по производительности.

С результатами можно ознакомиться в таблице.

Таблица

Результаты тестирования

Вершин в модели, тыс.	Среднее время работы метода при использовании разных алгоритмов оценки расстояния до вершин		
	Алгоритм Дейкстры, мс	Fast Marching, мс	Heat, мс
4.312	2	2	4
7.683	15	14	18
15.263	24	27	28
43.191	63	61	60
243.104	104	162	94
921.432	305	361	341
5691.1	1305	1484	1367

Можно сделать вывод, что применение методов поиска геодезического расстояния, в целом, дает немного худший по скорости

результат, чем использование алгоритма Дейкстры. Но проигрыш не столь существен, поэтому для достижения большей точности можно использовать и Fast Marching, и Heat.

Обратим внимание на то, что, согласно нашим результатам, использование Fast Marching для задачи понижения размерности 3D-модели выгоднее по скорости для моделей с меньшим количеством вершин. Для больших же объектов лучше применять Heat. Это наблюдение можно использовать при реализации алгоритма.

Список литературы

1. Garland, M. Surface Simplification Using Quadric Error Metrics / M. Garland, P.S. Heckbert // SIGGRAPH'97 : The 24th International Conference on Computer Graphics and Interactive Techniques. – 1997. – P. 209-216.
2. Aiello, R. Fast arbitrary geodesic computation on triangular meshes / R. Aiello. – Anno Accademico, 2013/2014. – 98 p.
3. Sethian, J. A. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science / J. A. Sethian. – Cambridge University Press, 1996. – 17 p.
4. Crane, K. Geodesics in Heat / K. Crane, C. Weischedel, M. Wardetzky. – Caltech, 2012. – 10 p.

Использование символьных вычислений и компьютерных алгоритмов при интегрировании алгебр Ли

В. В. Крутских

Студент бакалавриата

А. В. Лобода

Профессор

Введение

Задачей работы является изучение одного семейства 7-мерных вещественных алгебр Ли и, в частности, исследование орбит этих алгебр в 4-мерном комплексном пространстве.

Имея абстрактные представления алгебр, необходимо найти их реализации в виде алгебр векторных полей в J^4 , а затем проинтегрировать их.

В работе изучается семейство нильпотентных 7-мерных неразложимых алгебр Ли, зависящее от вещественного параметра. Полное описание такого рода семейств предложено М. П. Гонгом в [1]. Мы рассмотрим однопараметрическое семейство 147E, зависящее от параметра λ ($\lambda \neq 0, \lambda \neq 1$).

В связи с тем, что мы обсуждаем алгебры векторных полей, необходимо ввести несколько основных понятий.

Каждое рассматриваемое векторное поле в 4-мерном комплексном пространстве имеет вид

$$e_k = a_k(z) \frac{\partial}{\partial z_1} + b_k(z) \frac{\partial}{\partial z_2} + c_k(z) \frac{\partial}{\partial z_3} + d_k(z) \frac{\partial}{\partial z_4}, \quad k = \overline{1, 7}, \quad (1)$$

или в сокращенном виде

$$e_k : (a_k(z), b_k(z), c_k(z), d_k(z)), \quad k = \overline{1, 7}; \quad (2)$$

здесь $z = (z_1, z_2, z_3, z_4)$, а $a_k(z), b_k(z), c_k(z), d_k(z)$ – голоморфные функции своих аргументов.

Искомую орбиту алгебры обозначим $M = \{\Phi = 0\}$, тогда для ее поиска воспользуемся тождеством

$$\Re\{e_k(\Phi)|_M\} \equiv 0, \quad k = \overline{1, 7}. \quad (3)$$

2. Абелевы идеалы

Существенную роль при изучении алгебр Ли играют имеющиеся в таких алгебрах абелевы идеалы (см. [2]).

Обсуждаемая алгебра 147E имеет следующие содержательные коммутационные соотношения:

$$\begin{aligned} [e_1, e_2] &= e_4, \quad [e_1, e_3] = -e_6, \quad [e_1, e_5] = -e_7, \\ [e_2, e_3] &= e_5, \quad [e_2, e_6] = \lambda e_7, \quad [e_3, e_4] = (1 - \lambda)e_7. \end{aligned} \quad (4)$$

Коммутационные соотношения для остальных пар базисных элементов алгебры являются тривиальными, то есть $[e_j, e_k] = 0$.

Для описания реализации этой алгебры в [2] предложен алгоритм, который связан с поиском максимальных абелевых идеалов и дальнейшим упрощением базисных векторных полей найденного идеала. Для обсуждаемой алгебры найдено четыре 4-мерных идеала; наш дальнейший анализ связан с идеалом $I_4 = \langle e_3, e_5, e_6, e_7 \rangle$, одним из четырех. Используя технику работы [2], мы получаем три возможных реализации базисных полей абелева идеала.

$$\begin{aligned}
e_3 &: (1, 0, 0, 0) & (0, b_4(z_1), c_4(z_1), d_4(z_1)) & (0, 1, 0, 0) \\
e_5 &: (0, 1, 0, 0) & (0, 1, 0, 0) & (0, 0, c_5(z_1), d_5(z_1)) \\
e_6 &: (0, 0, 1, 0) & (0, 0, 1, 0) & (0, 0, 1, 0) \\
e_7 &: (0, 0, 0, 1) & (0, 0, 0, 1) & (0, 0, 0, 1)
\end{aligned} \tag{5}$$

Отметим, что для поиска абелевых идеалов был разработан отдельный алгоритм в пакете символьной математики Maple. Для начала необходимо представить коммутационные соотношения алгебры в виде 4-мерного массива, как в работе [1]. Далее было придумано обозначение для идеала, а именно – базис идеала представляется в виде вектора, содержащего индексы базисных элементов. Данный вектор проверяется на два условия:

1. Является ли выбранная подалгебра идеалом.
2. Является ли идеал абелевым.

3. Интегрирование алгебр Ли

С помощью алгоритмов, описанных в работе [2], была получена следующая реализация для семейства 147E:

$$\begin{aligned}
e_1 &: (1, 0, z_2, 0); \\
e_2 &: (\lambda C_5, B_4 z_1 + B_2, c(z_1, z_2), d(z_1, z_2, z_3)); \\
e_3 &: (0, 1, 0, 0); e_4 : (0, B_4, B_4 z_1 + C_4, (1 - \lambda)z_2 + D_4); \\
e_5 &: (0, 0, C_5, -z_1 + D_5); e_6 : (0, 0, 1, 0); e_7 : (0, 0, 0, 1);
\end{aligned} \tag{6}$$

где

$$\begin{aligned}
c(z_1, z_2) &= -C_5 z_2 + B_4 z_1^2 + (B_2 + C_4)z_1 + C_2, \\
d(z_1, z_2, z_3) &= -(-z_1 + D_5)z_2 - \lambda z_3 + D_4 z_1 + D_2,
\end{aligned} \tag{7}$$

а большими латинскими буквами обозначены комплексные константы.

Также на получившуюся реализацию имеется два условия

$$\begin{aligned}
\lambda C_5 B_4 + C_5 B_4 &= 0; \\
(1 - \lambda)B_2 + B_4 D_5 + \lambda C_4 &= 0.
\end{aligned} \tag{8}$$

Пользуясь тождеством (3), переходим от векторных полей к системе дифференциальных уравнений в частных производных.

$$\begin{aligned}
\frac{\partial F}{\partial x_1} + y_2 \frac{\partial F}{\partial y_3} &= 0; \\
\lambda c_{51} \frac{\partial F}{\partial x_1} + \lambda c_{52} \frac{\partial F}{\partial y_1} + (b_{31} y_1 + b_{32} x_1 + b_{22}) \frac{\partial F}{\partial y_2} + \\
+ (-c_{51} y_2 + b_{32} (x_1^2 - y_1^2) + 2b_{31} x_1 y_1 + b_{21} y_1 +
\end{aligned} \tag{9}$$

$$\begin{aligned}
& +b_{22}x_1 + c_{31}y_1 + c_{32}x_1 + c_{22})\frac{\partial F}{\partial y_3} = \\
& = d_{31}y_1 + d_{32}x_1 - b_{51}y_1 - \lambda y_3 + x_1y_2 + d_{22} ; \\
b_{32} \frac{\partial F}{\partial y_2} + (b_{31}y_1 + b_{32}x_1 + c_{32})\frac{\partial F}{\partial y_3} & = (1-\lambda)y_2 + d_{32} ; \\
c_{52} \frac{\partial F}{\partial y_3} & = -y_1 + d_{52} .
\end{aligned}$$

Можно отметить, что команда `pdsolve` пакета Maple , примененная ко всей этой системе, не дает ее решений. В связи с этим был использован комбинированный алгоритм решения такого рода систем.

4. Комбинированный алгоритм решения систем ДУЧП

Мы используем пошаговое решение (в пакете Maple) отдельных уравнений описанных систем с последующим упрощением и уточнением оставшихся уравнений (см. [3]). При таких уточнениях учитываются, например, условия на коэффициенты и другие моменты, трудно формализуемые в пакетах символьных вычислений. Для освоения и поиска нужных функций и процедур пакета авторы пользовались книгой [4].

Опыт показывает, что работу с комбинациями комплексных и вещественных переменных в пакете Maple удобнее переводить на использование только вещественных переменных, причем при постоянных проверках в ручном режиме текущих вычислений. Отметим, что вместо предварительного описания типов используемых переменных для этого проще использовать команду `evals`, не требующую таких описаний.

Перед интегрированием необходимо выписать имеющиеся условия (8) в пересчете на мнимые и вещественные части

$$\begin{aligned}
b_{31}c_{51}\lambda - b_{32}c_{52}\lambda + b_{31}c_{51} - b_{32}c_{52} & = 0 , \\
b_{31}c_{52}\lambda + b_{32}c_{51}\lambda + b_{31}c_{52} + b_{32}c_{51} & = 0 , \\
-b_{21}\lambda + b_{31}d_{51} - b_{32}d_{52} + c_{31}\lambda + b_{21} & = 0 , \\
-b_{22}\lambda + b_{31}d_{52} + b_{32}d_{51} + c_{32}\lambda + b_{22} & = 0 .
\end{aligned} \tag{10}$$

Далее поочередно (ручным выбором) решаются отдельные уравнения выписанной системы. Например, самое простое (последнее) из них легко решается как в ручном режиме, так и с помощью команды `pdsolve`:

$$F(x_1, y_1, y_2, y_3) = \frac{(d_{52} - y_1)y_3}{c_{52}} + {}_-\!F_1(x_1, y_1, y_2). \quad (11)$$

После подстановки полученного решения в оставшиеся уравнения (команда `pdetest`) в системе останется три уравнения, два из которых содержат по одной частной производной и являются относительно простыми.

Продолжая аналогичные действия, мы получим в итоге одноединственное уравнение. Его правая часть формально зависит от «лишней» переменной y_2 . Эта переменная удаляется из уравнения за счет использования связей между условиями (10) и выражениями для коэффициентов при переменной y_2 . Однако формализовать в пакете Maple такое исключение достаточно сложно, так что эта часть решения задачи реализуется «вручную»:

$$\begin{aligned} c_{52}\lambda \frac{d_-\!F_3(y_1)}{dy_1} = & -\frac{1}{b_{32}c_{52}}(-b_{31}^2d_{52}y_1^2 + b_{31}^2y_1^3 - b_{31}c_{52}d_{52}\lambda y_2 + \\ & + b_{31}c_{52}\lambda y_1y_2 - b_{32}^2d_{52}y_1^2 + b_{32}^2y_1^3 - b_{32}c_{51}d_{52}\lambda y_2 + b_{32}c_{51}\lambda y_1y_2 + \\ & + b_{21}b_{32}d_{52}y_1 - b_{21}b_{32}y_1^2 - b_{22}b_{31}d_{52}y_1 + b_{22}b_{31}y_1^2 - b_{22}c_{52}\lambda y_2 - \\ & - b_{31}c_{32}d_{52}y_1 + b_{31}c_{32}y_1^2 + b_{31}c_{52}d_{32}y_1 + b_{31}c_{52}y_1y_2 + b_{32}c_{31}d_{52}y_1 - \\ & - b_{32}c_{31}y_1^2 - b_{32}c_{51}d_{52}y_2 + b_{32}c_{51}y_1y_2 - b_{32}c_{52}d_{31}y_1 + b_{32}c_{52}d_{51}y_2 + \\ & + c_{32}c_{52}\lambda y_2 - b_{22}c_{32}d_{52} + b_{22}c_{32}y_1 + b_{22}c_{52}d_{32} + b_{22}c_{52}y_2 + \\ & + b_{32}c_{22}d_{52} - b_{32}c_{22}y_1 - b_{32}c_{52}d_{22}). \end{aligned} \quad (12)$$

В итоге последнее уравнение системы является обыкновенным дифференциальным уравнением первого порядка с разделяющимися переменными. Его решением является полиномиальная функция 4-й степени:

$$\begin{aligned} {}_-\!F_3(y_1) = & -\frac{1}{b_{32}c_{52}^2\lambda} \left(\left(\frac{1}{4}y_1^4b_{31}^2 + \frac{1}{4}y_1^4b_{32}^2 - \frac{1}{3}y_1^3b_{31}^2d_{52} - \right. \right. \\ & \left. \left. - \frac{1}{3}y_1^3b_{32}^2d_{52} - \frac{1}{3}y_1^3b_{21}b_{32} + \frac{1}{3}y_1^3b_{22}b_{31} + \frac{1}{3}y_1^3b_{31}c_{32} - \right. \right. \\ & \left. \left. - \frac{1}{3}y_1^3b_{32}c_{31} + \frac{1}{2}y_1^2b_{21}b_{32}d_{52} - \frac{1}{2}y_1^2b_{22}b_{31}d_{52} - \frac{1}{2}y_1^2b_{31}c_{32}d_{52} + \right. \right. \\ & \left. \left. + \frac{1}{2}y_1^2b_{31}c_{52}d_{32} + \frac{1}{2}y_1^2b_{32}c_{31}d_{52} - \frac{1}{2}y_1^2b_{32}c_{52}d_{31} + \frac{1}{2}y_1^2b_{22}c_{32} - \right. \right. \end{aligned} \quad (13)$$

$$-\frac{1}{2}y_1^2b_{32}c_{22} - b_{22}c_{32}d_{52}y_1 + b_{22}d_{32}c_{52}y_1 + \\ + c_{22}d_{52}b_{32}y_1 - d_{22}b_{32}c_{52}y_1) + C_1.$$

Общим решением системы (9) будет объединение всех решений, полученных на разных этапах интегрирования:

$$-F_1(x_1, y_1, y_2) = \frac{-y_2(d_{52} - y_1)x_1}{c_{52}} + -F_2(y_1, y_2). \\ F_2(y_1, y_2) = -\frac{\frac{1}{2}y_2^2c_{52}\lambda + b_{31}d_{52}y_1y_2 - b_{31}y_1^2y_2 -}{b_{32}c_{52}} \quad (14) \\ -\frac{\frac{1}{2}y_2^2c_{52} + c_{32}d_{52}y_2 - c_{32}y_1y_2 - d_{32}c_{52}y_2}{b_{32}c_{52}} + F_3(y_1).$$

На заключительном этапе поиска интегральных поверхностей полученные сложно устроенные функции удастся упростить (в ручном режиме) за счет использования допустимых замен координат и приведения полученных полиномиальных уравнений к канонической форме.

Часть коэффициентов таких полиномов при этом оказывается несущественной, однако не представляется возможным сказать заранее: какие именно коэффициенты окажутся несущественными?

Заключение

Реализованный с использованием пакета Maple алгоритм решения систем уравнений в частных производных позволил получить их полные решения.

Приведение полученного полиномиального решения к канонической форме дает полное решение поставленной задачи в виде следующего уравнения интегральной поверхности для рассматриваемого семейства алгебр Ли (в наиболее интересной несферической части):

$$y_4 = y_1y_3 + y_2^2 + x_1y_1y_2 + Dy_1^4, \quad D \neq \frac{1}{12}. \quad (15)$$

Достигнутый результат позволяет рассчитывать на возможность использования компьютерных технологий при получении описаний более широкого круга объектов в обсуждаемой многомерной математической задаче.

Список литературы

1. Gong, M.-P. Classification of Nilpotent Lie Algebras of Dimension 7 (Over Algebraically Closed Fields and \mathbb{R}) / M.-P. Gong. – University of Waterloo, 1998. – 165 p.
2. Loboda, A. V. On the Orbits of Nilpotent 7-dimensional Lie Algebras in 4-dimensional Complex Space / A. V. Loboda, R. S. Akopyan, V. V. Krutskikh // J. Sib. Fed. Univ. Math. Phys. – 2020. – 13(3). – P. 360-372.
3. Крутских, В. В. Алгебры Ли нелинейных векторных полей / В. В. Крутских, А. В. Лобода // Сборник студенческих научных работ факультета компьютерных наук ВГУ. – 2019. – Вып. 13. – Ч. 1. – С. 105-110.
4. Дьяконов, В. П. Maple 9.5/10 в математике, физике и образовании / В. П. Дьяконов. – М. : СОЛОН-Пресс, 2006. – 720 с.

Система учета посещаемости на примере ФКН

В. К. Кушнеренко

Студент бакалавриата

А. И. Чекмарёв

Ассистент

Введение

В современном мире практически во всех учебных заведениях широко распространены различные документы по учету посещаемости. Эти заведения используют те данные, которые предоставляются старостами каждую неделю, что не способствует получению своевременной информации о посещаемости. Документы по учету посещаемости - это документы, в которых изложено отсутствие или присутствие человека на занятиях.

Учет посещаемости требует большое количество затрат: сопровождается значительным объёмом документов, отнимает много времени.

Принимая во внимание вышеописанные факты, мы можем сформулировать ряд конкретных проблем:

- Необходимость затрат большого количества времени на заполнение документов;
- Неструктурированность документов по учету посещаемости;

- Старосты не всегда добросовестно заполняют посещаемость;
- Необходимо хранить все документы по учету в бумажном виде;
- Сложность в просмотре статистики по каждому студенту;

Отдельно рассмотрим учет посещаемости через нашу систему.

Разработка учета посещаемости позволит решить указанные проблемы процесса его проведения, который используется в настоящее время, а также сократить затраты времени на заполнение документов по учету.

1. Постановка задачи

Целью данной работы является разработка и внедрение учета посещаемости.

Конечными пользователями системы будут студенты, которые будут иметь возможность просматривать статистику по посещениям, и преподаватели, которые смогут также следить за статистикой и заполнять данные.

К разрабатываемому продукту предъявляются следующие требования:

- Удобный просмотр и редактирование учета посещаемости;
- Удобное взаимодействие с элементами системы;
- Работа под несколькими ролями в системе;
- Возможность работать с данными, связанными с посещаемостью;

Для создания качественного решения необходимо провести анализ существующих решений.

2. Анализ

Анализ предметной области

Процесс учета включает в себя комплекс мероприятий, направленных на формирование документов о посещаемости студентов и сбора различной актуальной информации.

При осуществлении учета посещаемости преподаватель получает список/журнал посещаемости. Затем он заходит в журнал посещаемости и отмечает не пришедших на занятие студентов. После этого он может продолжать занятие. Имеется возможность изменять данные студентов, дисциплин, добавлять новые группы. Также можно провести поиск по конкретному студенту.

В данной работе будет проведён детальный разбор функционирования системы.

Анализ задачи

Рассмотрим задачу экспорта данных о посещаемости из файла. В настоящий момент имеются файлы с записями, общая структура которых представлена в таблице.

Таблица

Упрощенное представление списка студентов для посещаемости.

ФИО студента/Дата проведения занятия	01. 05. 02	02.05.02	03.05.02
Беркутов И. С.	+	-	+
Меркулов С. В.	-	+	+
Петров П. В.	+	+	+
Сидоров В. И.	+	-	-

В 1 ячейке хранится параметр, обозначающий то что находится в ячейках внизу (ФИО студентов) и справа (даты занятия), в остальных ячейках находятся отметки - плюс или минус в зависимости от того был ли студент на занятии или нет.

Соответственно, экспорт в PDF данных студентов по посещаемости должен быть разработан с учетом структуры файла с формализованными требованиями.

Теперь проанализируем задачу разработки системы.

Для работы в системе предусмотрены разные роли. Роли могут принимать следующие значения:

- Студент
- Преподаватель
- Староста
- Администратор

Отметки посещаемости могут иметь свои статусы:

- Плюс - студент был на занятии
- Минус - студент не был на занятии

Система должна упрощать процесс своевременного получения данных, а также отслеживание статистики посещаемости, с учетом текущих данных.

Существующие решения

В ходе проведения анализа были рассмотрены следующие готовые решения по учету посещаемости, в скобках приведены ссылки на официальные сайты существующих аналогов:

- Moodle (<https://edu.vsu.ru/>) Система позволяет не только отслеживать посещаемость, но и выставять оценки за курсы, общаться

с преподавателями онлайн. Внешний вид сайта, представляющего систему представлен на рис. 1.

– iTeacherBook – журнал преподавателя (https://ipadstory.ru/iteacherbook-zhurnal-prepodavatelya.html) – рис. 2. Этот сайт позволяет хранить информацию о студентах, их посещаемости, оценках и других различных данных.



Рис. 1. Официальный сайт системы Moodle

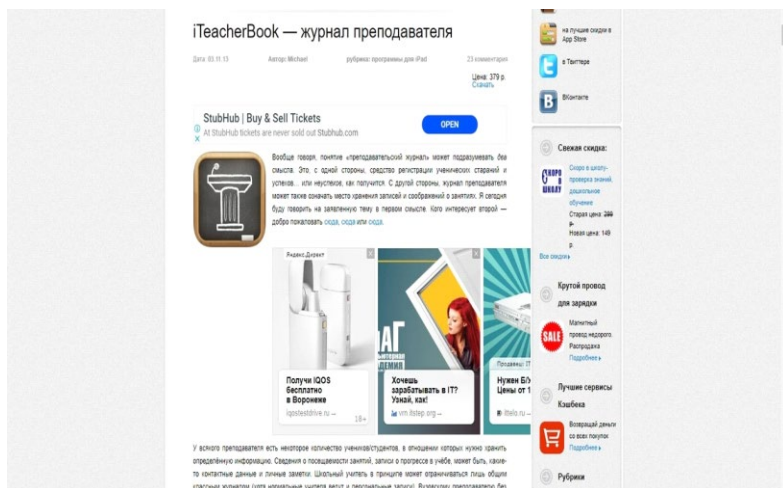


Рис. 2. Официальный сайт системы iTeacherBook

3. Реализация

Средства реализации

Для реализации проекта были выбраны следующие технологии и программные средства:

- Язык программирования Python 3.7 - обладает простотой и функциональностью, широко используется для разработки веб-приложений.

- Библиотека Flask [1] из языка программирования Python – обеспечивает быстрое и легкое создание веб-приложений и различных необходимых для его функционирования модулей.

- Библиотека Flask-Login - для реализации авторизации и аутентификации в системе по ролям.

- Шаблонизатор Jinja [2] – для обработки данных, поступающих на HTML-страницу.

- Библиотека SQLAlchemy [3] - для создания прослойки в виде классов, связанных с БД.

- Система контроля версий Git – обеспечивает возможность сохранения состояний проекта, получения текущей версии проекта и извлечение старых версий. Более того, система позволяет отслеживать этапы разработки решения.

Реализация классов

В результате анализа было принято решение о разработке системы, описание которой приводится в данной главе.

Структура проекта имеет вид:

- Файл `app.py` – содержит в себе все маршруты веб-приложения. Также здесь идет печать в PDF и авторизация.
- Файл `app_config.py` – конфигурация и настройки всего проекта.
- Файл `forms.py` – предоставляет все формы, необходимые для функционирования приложения.
- Файл `model.py` – представляет собой модели БД в виде их классов.
- `templates` – представляет собой папку с шаблонами html-страничек.

Диаграмма вариантов использования

При разработке системы составлена диаграмма вариантов использования, изображенная на рис. 3.

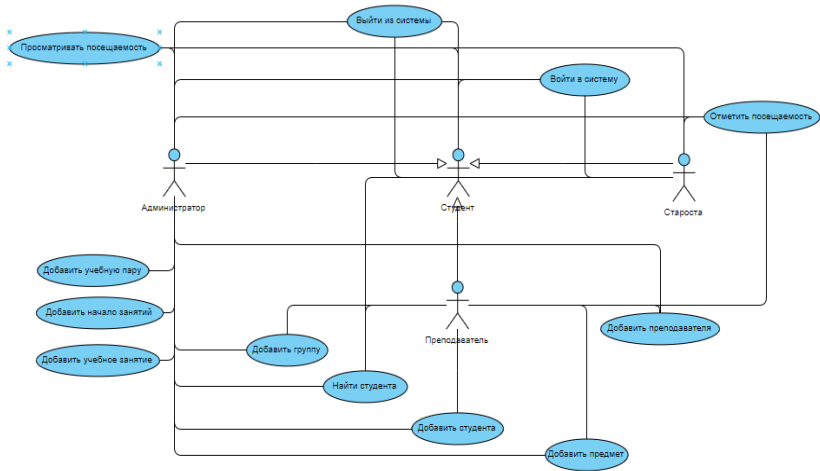


Рис. 3. Диаграмма вариантов использования

Диаграмма классов

При разработке системы составлена диаграмма классов, изображенная на рис. 4.

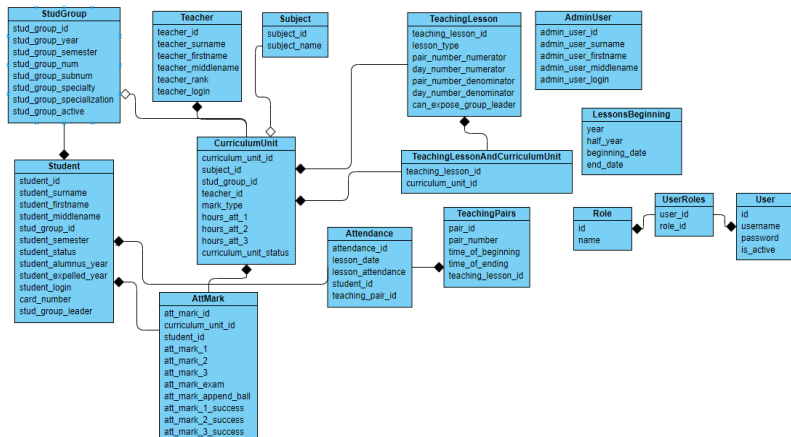


Рис. 4. Диаграмма классов для системы

Схема данных

При разработке системы составлена схема данных, изображенная на рис. 5.

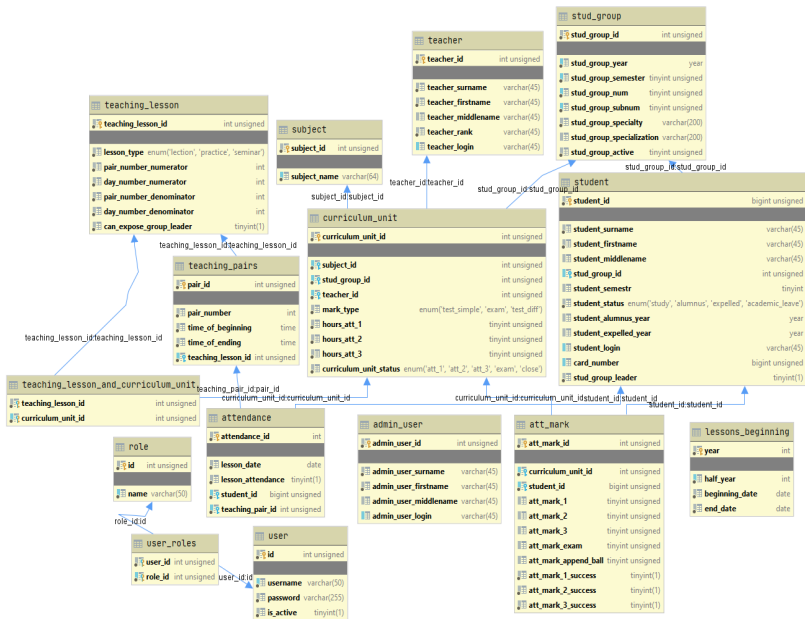


Рис. 5. Схема данных системы

Stud_group – таблица, содержащая группы студентов.

Student – таблица, содержащая в себе данные конкретного студента.

Teacher – таблица, содержащая данные о преподавателях в учебном заведении.

Curriculum_unit – таблица, содержащая единицы учебного плана.

Subject – таблица, содержащая сведения о предметах в учебном заведении.

Teaching_lesson – таблица, содержащая сведения о занятиях в учебном заведении.

Teaching_lesson_and_curriculum_unit – связующая таблица между единицами учебного плана и занятиями в учебном заведении.

Attendance – таблица, содержащая сведения о посещаемости студента.

Att_mark – таблица, содержащая в себе аттестационные результаты студента.

Teaching_pair – таблица, содержащая в себе учебные пары на определенные дни.

Lessons_beginning – вспомогательная таблица для учета по первой неделе числителя/знаменателя.

User – таблица, содержащая информацию о пользователях в системе.

Role – таблица, содержащая информацию о ролях пользователя.

User_roles – связующая таблица между пользователями и их ролями.

Заключение

В ходе выполнения работы были проанализированы существующие решения для учета посещаемости, в результате анализа был получен вывод, что ни одно из них полностью не соответствует требованиям и не обладает необходимым функционалом.

Результатом выполнения работы является набор инструментов, позволяющий управлять учетом посещаемости и возможность просмотра данных о ней.

Список литературы

Сетевые ресурсы:

1. Документация по Flask [Электронный ресурс]. – Режим доступа : <http://flask.pocoo.org/docs/1.0/>
2. Документация по Jinja [Электронный ресурс]. – Режим доступа : <http://jinja.pocoo.org/docs/2.10/>
3. Документация по SQLAlchemy [Электронный ресурс]. – Режим доступа : <https://www.sqlalchemy.org/>

Электронный датчик расхода воды

А. В. Москаленко

Студент бакалавриата

Н. С. Быканов

Студент бакалавриата

С. А. Зуев

Доцент

Введение

Использование микропроцессоров Arduino находит всё больше и больше применений в различных сферах деятельности человека. Микропроцессоры Arduino позволяют сделать многие обыденные дела

гораздо проще, автоматизировав различные процессы, за которыми был необходим присмотр человека. Эти устройства популярны среди любителей электроники, так как довольно просты в использовании и имеют довольно широкий круг пользователей, которые постоянно поддерживают начинающих на форумах и интернет-площадках.

Благодаря многочисленным преимуществам Arduino, одним из которых является малое энергопотребление, представляется возможным создание устройств, способных вести автоматический подсчёт величин, характеризующих расход человеком всевозможных ресурсов. В данной работе будет рассмотрено применение микропроцессора и датчика расхода воды [1] для контроля истечения ресурса встраиваемого фильтра для воды с использованием простого интерфейса, выводимого на экран и светового оповещения в случае истечения ресурса фильтра.

1. Цель работы

Цель данной работы – сконструировать и исследовать на практике работу действующей модели электронного счётчика расхода воды при помощи микропроцессора Arduino.

Требования к проекту:

- Точное измерение расхода воды с допустимой погрешностью 0,5л на 100 литров израсходованной воды.
- Простое управление микропроцессором, индивидуальная настройка счётчика в зависимости от ресурса фильтра
- Устойчивость системы и сохранение данных, даже в случае отключения питания счётчика
- Система оповещения пользователя в случае израсходования ресурса фильтра, сигнализирующая пользователя о необходимости смены фильтра

2. Датчики и дополнительные механизмы

- Микропроцессор Arduino Nano 3.0.
- LCD дисплей MT-16S2H.
- Датчик расхода воды YF-S201.
- Трубка необходимого диаметра для подключения к фильтру.

3. Схема работы

Датчик расхода воды при пропускании через себя некоторого количества жидкости будет подавать аналоговый сигнал, который преобразуется в цифровой и воспринимается микропроцессором как изменение количества расходуемой воды. Эти данные (в литрах) будут сохраняться благодаря EEPROM памяти [2], таким образом минимизировав получение возможной погрешности до значения,

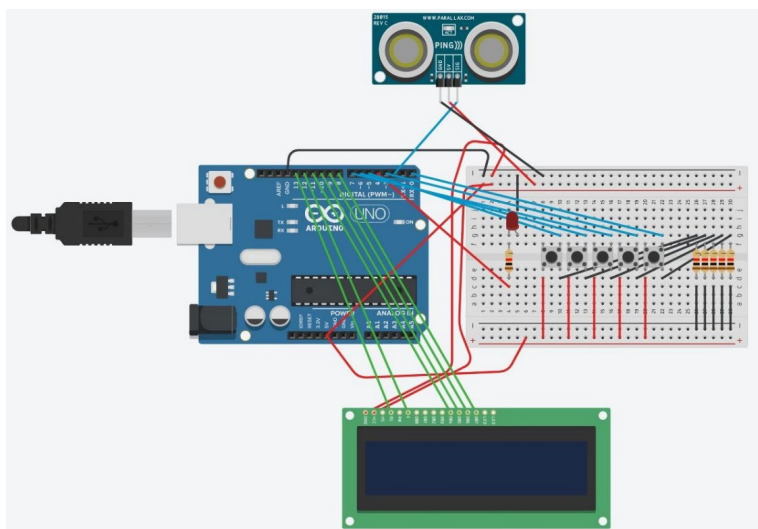
меньше чем 1 литр (так как при отключении питания будут обнуляться только миллилитры). Устройство оснащено дисплеем, на который выводится значение объёма израсходованной воды и светодиодом, который будет подавать световой сигнал в случае израсходования заданного ресурса [3]. Также на устройстве установлены кнопки, позволяющие устанавливать ресурс фильтра, проводить калибровку датчика расхода воды путём пропускания через него жидкости объёмом 1000мл, включать/выключать подсветку дисплея и обнулять значение счётчика [4].

4. Программная часть

Для управления работой датчиков и микропроцессора использовался язык программирования C++.

Ссылка на исходный код находится в [5].

Схему соединения можно увидеть на рис. 1, а на рис. 2 можно наблюдать устройство в штатном режиме работы.



Ультразвуковой датчик на схеме – датчик расхода воды

Рис. 1. Соединение.

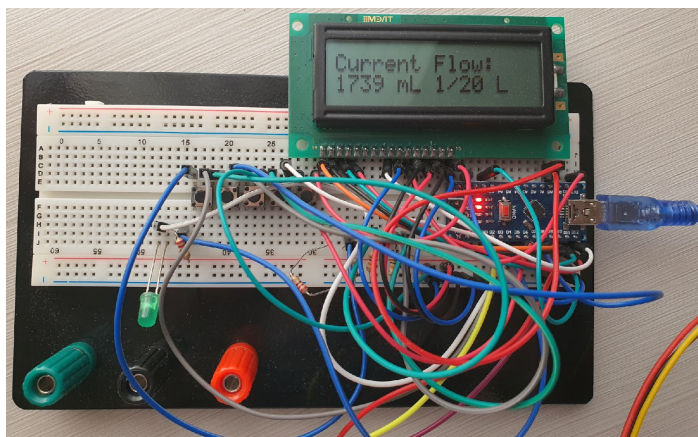


Рис. 2. Собранное устройство в рабочем режиме

5. Практическое применение

Данное устройство легко настраиваемо и может быть интегрируемо в любую систему встроенной фильтрации воды за счёт своего небольшого размера, что можно заметить на рис. 3. Это позволяет настроить устройство в зависимости от ресурса вашего фильтра и отслеживать его, вовремя оповещая пользователя о том, что необходимо сменить картридж фильтра. Благодаря этому пользователь будет пить всегда чистую воду и не будет обманут продавцами сменных картриджей, которые пытаются продать их клиенту задолго до истечения ресурса предыдущих.

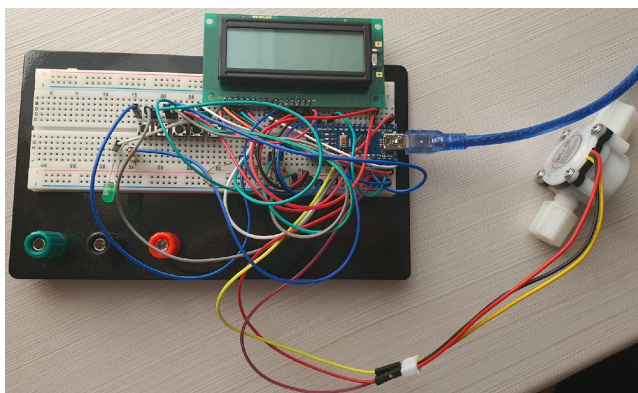


Рис. 3. Собранное устройство в выключенном состоянии

Заключение

В ходе выполнения данной работы мы сконструировали удобный и настраиваемый электронный счётчик воды, устойчивый к отключениям питания и сохраняющий значения расходуемого ресурса в EEPROM памяти. Полученное устройство обладает довольно низкой погрешностью и включает в себя световую систему оповещения пользователя. В дальнейшем предполагается изготовить корпус из пластика на 3-D принтере и добавить блок передачи файлов по Wi-Fi для включения прибора в систему «Умный дом».

Список литературы

1. Датчик потока воды [Электронный ресурс]. – Режим доступа : http://digitrode.ru/computing-devices/mcu_cpu/1806-arduino-i-datchik-rashoda-vody.html
2. Arduino – работа с EEPROM памятью [Электронный ресурс]. – Режим доступа : <https://alexgyver.ru/lessons/eprom>
3. Соммер, У. Программирование микроконтроллерных плат Arduino/Freeduino / У. Соммер; пер. с нем. – 2-е изд., перераб. и доп. – СПб : БХВ-Петербург, 2016. – 256 с.
4. Петин, В. А. Проекты с использованием контроллера Arduino / В. А. Петин. – СПб : БХВ-Петербург, 2015. – 448с.
5. Исходный код [Электронный ресурс]. – Режим доступа : https://github.com/itsremor/vsu_science/blob/master/watermeter.ino

Реализация анонимного ретранслятора IEEE 802.11

Р. А. Никулин

Студент бакалавриата

А. А. Головкин

Ассистент

Введение

Актуальной задачей при эксплуатации беспроводных сетей является расширение зоны покрытия беспроводных устройств. Для решения этой задачи в рамках Wi-Fi-сетей традиционно используются ретрансляторы (репитеры), построенные на базе беспроводных точек доступа. Особенностью этого решения является необходимость идентификации ретранслятора в существующей сети, что усложняет

использование ретрансляторов в корпоративных Wi-Fi-сетях и увеличивает стоимость оборудования.

Одним из вариантов расширения покрытия беспроводных сетей является использование анонимных ретрансляторов. Для функционирования им не требуется прямое авторизованное подключение к ретранслируемой сети.

1. Виды и структура Wi-Fi трафика

Для решения задачи выборочной трансляции кадров необходимо проанализировать их общую структуру.

Стандарт 802.11 определяет три класса кадров, передаваемых по радиоканалу: информационные, служебные и управляющие. Все они имеют заголовки с множеством полей, используемых подуровнем MAC. Кроме того, есть поля, используемые физическим уровнем, но они в основном относятся к методам модуляции, что выходит за рамки решаемой задачи [1].

Информационный класс кадров:

Информационный кадр – контейнер для передаваемой пользовательской информации в инфокоммуникационных сетях.

Управляющий класс кадров:

Кадры данного класса позволяют устанавливать и поддерживать коммуникации в сети, их анализ и обработка составляют основу работы анонимного ретранслятора. Рассмотрим некоторые из них подробнее:

Кадры подкласса Beacon [2] представляют собой периодические широковещательные сообщения точки доступа (например, каждые 100 миллисекунд). Они сообщают клиентам о присутствии точки доступа и несут системные параметры, такие как идентификатор, время, интервал до следующего маяка и настройки безопасности. На рис. 1 представлена структура управляющего кадра подкласса Beacon [3].

Кадры подкласса Probe Request [3] используются мобильными станциями для сканирования радиоэфира на присутствие точек доступа. На рис. 2 представлена структура управляющего кадра подкласса Probe Request.

Probe Response. Кадры данного подкласса являются ответом точки доступа на кадры подкласса Probe Request при условии удовлетворения точки доступа параметрам SSID, Supported Rates и Extended Supported. На рис. 3 представлена структура управляющего кадра подкласса Probe Response [3].

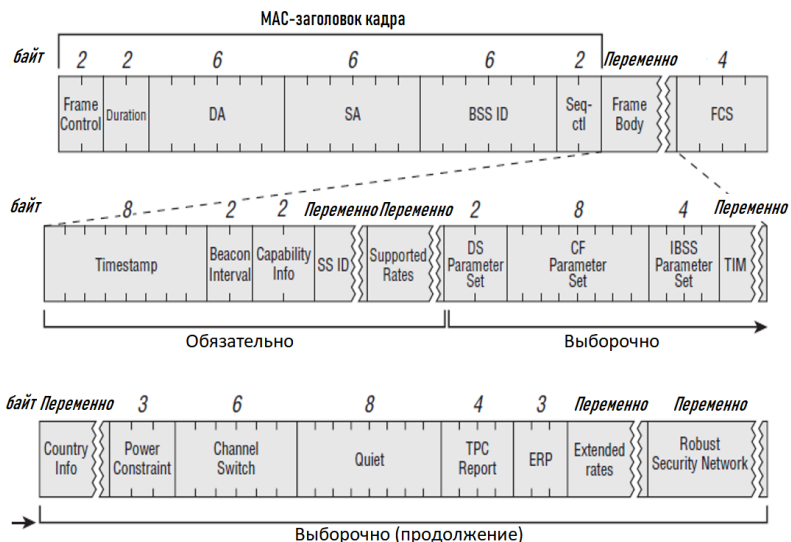


Рис. 1. Структура кадра подкласса Beacon

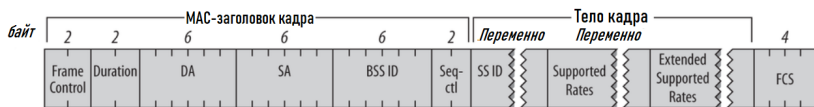


Рис. 2. Структура кадра подкласса Probe Request

2. Алгоритм работы анонимного ретранслятора

Как уже отмечалось ранее, анонимный ретранслятор позволяет расширить покрытие беспроводной сети без прямого соединения с ней. Среда передачи сканируется на предмет наличия кадров, принадлежащих транслируемой сети, а именно:

Происходит захват всех кадров в среде передачи;

Анализируется MAC-заголовок кадра;

Кадр ретранслируется в среду передачи, если в его заголовке фигурирует BSSID транслируемой сети.

Для анализа принадлежности кадра к конкретной сети необходимо узнать BSSID точки доступа этой сети. Также, в целях удобства должна существовать возможность установления соответствия между SSID и BSSID точки доступа. Управляющий класс кадров содержит в себе подклассы, удовлетворяющие указанным требованиям.

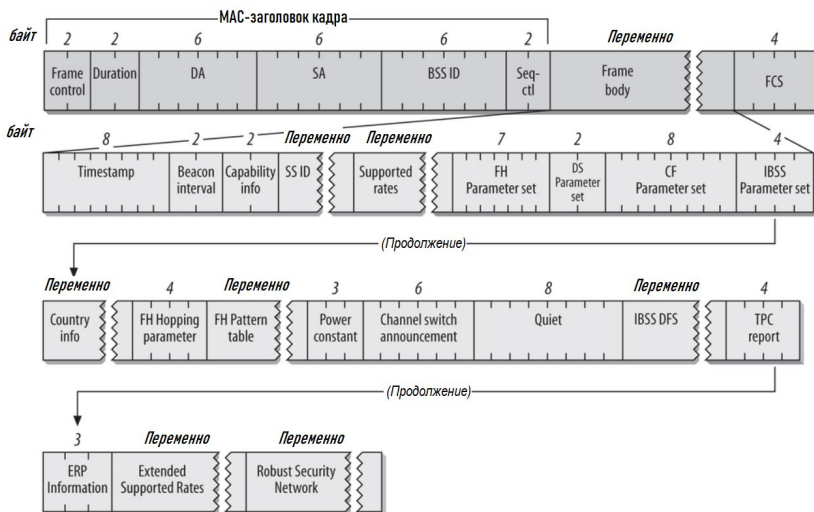


Рис. 3. Структура кадра подкласса Probe Response

С помощью кадров подкласса Weacon можно реализовать пассивное сканирование: прослушивание радиоэффира на предмет наличия кадров подкласса Weacon с последующим извлечением информации о SSID и BSSID из них.

С помощью кадров подкласса Probe Request и Probe Response можно реализовать активное сканирование, которое заключается в отправке запросов, представленных в формате кадра подтипа Probe Request, точке доступа с помощью оконечного устройства и дальнейший анализ ответа, предоставленного точкой доступа, в формате кадра подтипа Probe Response.

В целях снижения энергопотребления анонимного ретранслятора и снижения нагрузки на сеть, был выбран подход пассивного сканирования, то есть анализ кадров подкласса Weacon.

3. Реализация анонимного ретранслятора

Диаграммы состояний программной реализации анонимного ретранслятора представлены на рис. 4 и рис. 5.

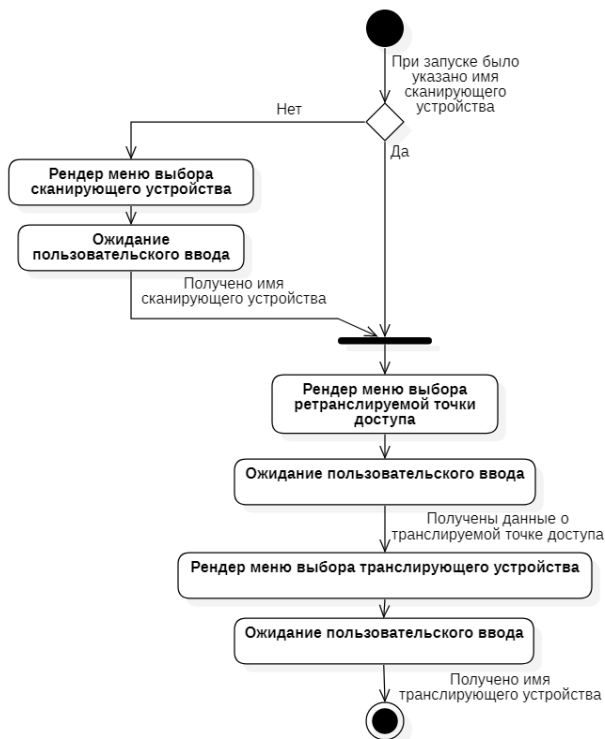


Рис. 4. Диаграмма состояний ввода пользователем необходимых данных

Прототип ретранслятора построен на базе ОС Linux, программная часть реализована на языке C++ [4-7]. Для захвата и анализа трафика используется библиотека LibPcap [8]. Аппаратная часть – ПК и Wi-Fi адаптеры диапазона 2,4 ГГц. Для проверки корректности работы алгоритма использовалось ПО WireShark.

Заключение

В статье предложен алгоритм работы анонимного Wi-Fi ретранслятора, в основе которого лежит анализ структуры управляющих кадров уровня L2 сетевой модели OSI. Предложенный алгоритм реализован в программно-аппаратном комплексе на базе ОС Linux. Разработанный прототип позволяет производить перевод сетевого адаптера в режим монитора, соотносить SSID и BSSID ретранслируемой точки доступа, ретранслировать кадры.

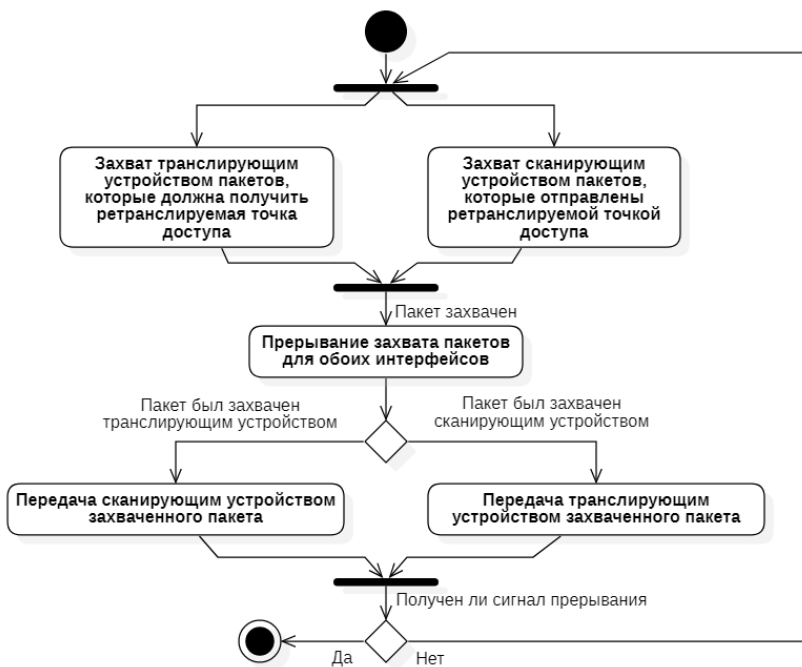


Рис. 5. Диаграмма состояний ретрансляции пакетов

Список литературы

1. Таненбаум, Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – 5-е изд. – М. : Питер, 2016. – 960 с.
2. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – 6-е изд. – М. : Питер, 2016. – 996 с.
3. Gast, M. 802.11 Wireless Networks: The Definitive Guide / M. Gast – 2nd edition. – Sebastopol : O'Reilly Media, 2009. – 656 p.
4. iee80211.h File Reference [Электронный ресурс] : информационный портал – Режим доступа : http://dox.ipxe.org/ieee80211_8h_source.html
5. Страуструп, Б. The C++ Programming Language / Б. Страуструп. – 2-е изд. – М. : Addison-Wesley Professional, 2013. – 1030 с.
6. Митчелл, М. Программирование для LINUX. Профессиональный подход / М. Митчелл, Д. Оулдем, А. Самьюэл. – 2-е изд. – М. : Вильямс, 2003. – 279 с.

7. UNIX: разработка сетевых приложений / Стивенс У. [и др.]. – 6-е изд. – М. : Питер, 2007. – 1051 с.

8. Tim Carstens. – Programming with Pcap [Электронный ресурс] : информационный портал. – Режим доступа : <https://www.tcpdump.org/pcap.html>

Разработка прототипа приложения для построения блок-схем по коду алгоритмов

Е. И. Олейник

Студент бакалавриата

Д. И. Соломатин

Старший преподаватель

Введение

Блок-схемы отображают принцип работы определённых алгоритмов, что даёт полное понимание происходящих процессов. Их достоинство в том, что они универсальны и описывают действие алгоритма так, как его можно реализовать на любом языке программирования. Поэтому для человека, прежде не имевшего никакого отношения к программированию, блок-схемы – это полезный инструмент в его изучении. Особенно это актуально для школьников, начинающих осваивать программирование. Поэтому практически во всех школьных учебниках по информатике изучение основных принципов и алгоритмов начинается с разбора простеньких задач на псевдо-коде и прилагающихся к ним блок-схем, которые обучающиеся потом начинают строить сами.

Для самостоятельного построения блок-схем существует огромное количество приложений, позволяющих создать схему нужного формата и оформления. Но для освоения такой программы и, собственно, самого создания диаграммы требуется время. Поэтому актуальными являются программы, которые автоматически генерируют блок-схему и предоставляют уже готовое изображение.

Но выбор такого вида приложений не настолько обширен, как хотелось бы. В основном такие программы предназначены для конкретных языков программирования, с которыми новичок может быть не знаком. А вариантов, поддерживающих псевдо-код, остаются

единицы. И зачастую, они выдают диаграммы, не соответствующие требованиям ГОСТа, которых придерживаются школьные учебники и преподаватели. К тому же, такие ресурсы в большинстве находятся в англоязычном сегменте интернета и являются платными.

1. Постановка задачи

Целью данной работы является создание прототипа приложения, предоставляющего возможность на основе введённого псевдо-кода получить изображение блок-схемы, соответствующую основным требованиям ГОСТа.

В качестве псевдо-кода использовался так называемый «школьный алгоритмический язык», использующий понятные школьнику слова на русском языке, широко применяющийся в учебниках по информатике и программированию.

Что касается требований к оформлению самих диаграмм, за их основу была взята единая система программной документации (ЕСПД), частью которой является Государственный стандарт – ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем», где описано, в том числе, и требования к блок-схемам [1]. Но так же пришлось учитывать и общие правила, применимые к представлению тех диаграмм, что приведены в школьных учебниках. В общих чертах должно соблюдаться:

- черно-белое оформление;
- прямые стрелки;
- относительно одинаковые размеры блоков;
- использование специальных фигур, соответствующих этапам работы алгоритма.

2. Предлагаемый подход

Для реализации программной части был выбран язык Python, а для автоматической визуализации графов – пакет утилит Graphviz, так как это доступная и известная библиотека, с которой можно работать на большинстве распространённых языков, в том числе на Python, и позволяющая визуализировать любую схему, которую можно представить в виде графа.

В общих чертах процесс преобразования кода в блок-схему представлен на рис. 1 и выглядит следующим образом: сперва синтаксический анализатор для псевдо-кода строит AST-дерево, затем анализируются его узлы и происходит интерпретация их значений в язык описания графов DOT, который записывается в файл с расширением .dot и в итоге через Graphviz преобразовывается в изображение с готовой диаграммой.

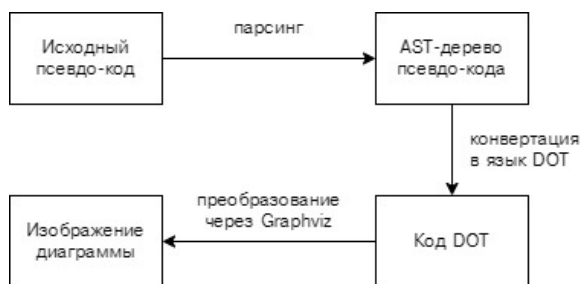


Рис. 1. Преобразование псевдо-кода в блок-схему

Рассмотрим эти шаги на каком-либо конкретном алгоритме. Например, на сортировке методом пузырька (см. листинг 1).

Листинг 1

Пример входной программы на псевдо-коде

```

алг Сортировка
цел N, arr[1:N], i, j, tmp
нач
ввод N, arr[1:N]
нц для i от 1 до N-1
  нц для j от i+1 до N
    если arr[j] > arr[i]
      то tmp=arr[j]
      arr[j]=arr[i]
      arr[i]=tmp
    все
  кц
кц
вывод arr[1:N]
кон
  
```

После парсинга в полученном AST-дереве почти все внутренние узлы представляют собой отдельные ключевые слова из массива токенов, определяемых грамматикой псевдо-кода, а большинство краевых узлов являются выражениями-идентификаторами, определяемыми пользователем, которые будут заполнять блоки диаграммы (см. листинг 2).

Далее происходит рекурсивный обход дерева, во время которого постепенно генерируется DOT-код.

Чтобы сделать граф похожим на блок-схему, задаём в начале его описания настройки для типов линий и размера всех узлов [2] (см. листинг 3)

Пример части AST-дерева

```

...
| если
|   arr[j] > arr[i]
|   то
|   | tmp=arr[j]
|   | arr[j]=arr[i]
|   | arr[i]=tmp
|   все
...

```

Настройка характеристик графа

```

graph [splines=ortho]
node[height = 0.5, width = 2]

```

Для нумерации всех появляющихся узлов в DOT-коде и дальнейшей возможности их соединения рёбрами используется переменная `end_node`, увеличивающаяся на 1 при каждом появлении нового узла, и таким образом, хранящая в себе номер последнего узла, требующего соединения. При возникновении новой ветки рекурсии обхода дерева, создаётся новый экземпляр этой переменной, значение которой будет меняться уже независимо от предыдущей. Таким образом, будет возможно соединение узлов на разных уровнях дерева.

Помимо узлов, представляющих собой сами элементы блок-схемы, использовались специальные «невидимые» узлы [3] (`n[shape=point, style = invis, height = 0, width = 0];`), служащие для обеспечения слияния рёбер в нужных местах и для управления поворотом стрелки в некоторых проблемных случаях, например при возвращении в начало цикла (номера узлов для поворота пропускаются переменной `end_node`, так как к ним не подсоединяются рёбра).

Код для описания элементов графа генерируется в зависимости от значения (ключевого слова) текущего узла дерева и содержания его последующих дочерних узлов. Так например, для ключевого слова «если» сперва происходит проверка на наличие ветки «иначе» (проверка значения третьего дочернего узла). Далее генерируется строка кода с описанием нового узла `<n6[label = "arr[j] > arr[i]", shape = diamond];>`, где в `label` вставляется соответствующая информация из первого дочернего для «если» элемента, то есть, `<arr[j] > arr[i]>`. А в `shape` подставляется соответствующий блоку условия тип фигуры (в DOT есть все необходимые для блок-схемы формы). Затем создаётся строка, описывающая ребро, соединяющее новый узел с тем, на который

указывает `end_node` (в данном случае «n4 -> n6;»), а также появляется невидимый узел `n7`, из которого пойдёт разветвление (`n7-> n8[label = "да"];... n7 -> n11[label = "нет"];`). Поэтому значение `end_node` увеличится сразу на 2. В итоге происходит рекурсивный вызов функции для второго и третьего (если он имеется) дочернего узла, для обработки веток «то» и «иначе».

3. Пример

После обхода всего дерева получаем файл `.dot` со сгенерированным DOT-кодом (см. листинг 4), который затем обрабатывается в Graphviz.

Листинг 4

Пример сгенерированного DOT-кода

```
digraph G {
graph [splines=ortho]
node[height = 0.5, width = 2]
n0[label = "начало"];
n1[label = "ввод N, arr[1:N]", shape=parallelogram];
n2[shape=point, style = invis, height = 0, width = 0];
n3[label = "i=1, N-1", shape = hexagon];
n4[label = "j=i+1, N", shape = hexagon];
n5[shape=point, style = invis, height = 0, width = 0];
n6[label = "arr[j] > arr[i]", shape = diamond];
n7[shape=point, style = invis, height = 0, width = 0];
n8[label = "tmp=arr[j]", shape = box];
n9[label = "arr[j]=arr[i]", shape = box];
n10[label = "arr[i]=tmp", shape = box];
n11[shape=point, style = invis, height = 0, width = 0];
n12[label = "конец"];

{rank = same; n4 n5};
{rank = same; n10 n11};

n0 -> n1;
n1 -> n2[dir=none];
n2 -> n3;
n3 -> n4;
n4 -> n6;
n6 -> n7[dir=none];
n7 -> n8[label = "да"];
n8 -> n9;
n9 -> n10;
n10 -> n11[dir=none];
n7 -> n11[label = "нет"];
n11 -> n4;
n4 -> n5[dir=none];
n5 -> n2;
n3 -> n12;}
```

Получить изображение можно вызвав команду
dot -Tpng input.dot > output.png
На рис. 2 изображён результат работы библиотеки.

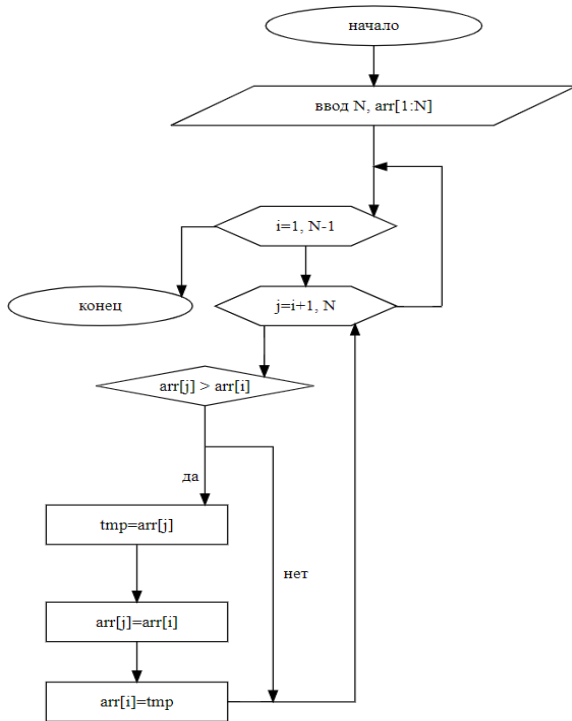


Рис. 2. Сгенерированная блок-схема для введённого алгоритма

Заключение

В итоге был разработан прототип приложения, удовлетворяющий основным требованиям поставленной задачи. Но в целом остались моменты, требующие доработок. В особенности, оформление диаграмм. Желательно, чтобы они стали выглядеть более симметрично и больше удовлетворяли требованиям ГОСТа. В дальнейшем можно будет добавлять обработку и для других языков программирования с более свободным синтаксисом и специальными командами, чтобы не только код мог находиться в блоках, но и какие-либо описания для изображения алгоритмов в общих чертах.

Список литературы

1. ГОСТ 19.701–90 (ИСО 5807–85) Единая система программной документации [Электронный ресурс]. – Режим доступа : <http://docs.cntd.ru/document/9041994>
2. Использование Graphviz для построения блок-схем [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/post/337078/>
3. Описание графов на языке DOT [Электронный ресурс]. – Режим доступа : <http://dkhramov.dp.ua/opisanie-grafov-na-iazyke-dot#.XsLH0WgzZPY>

Прогнозирование размера страховой медицинской выплаты пациентам методами машинного обучения

А. С. Палади

Студент бакалавриата

А. В. Копытин

Доцент

Введение

В России в настоящее время действуют две страховых системы – обязательное медицинское страхование (ОМС) и добровольное медицинское страхование (ДМС). На страховку ОМС может рассчитывать каждый россиянин. Она гарантирует бесплатный минимум диагностики и лечения в рамках государственной программы. В свою очередь, страховка ДМС оформляется гражданином на добровольной основе у частных коммерческих организаций и дополняет минимум до оптимума качественными медицинскими услугами в условиях улучшенного сервиса, а также обеспечивает материальную поддержку в случае получения травм.

Чтобы получить свою прибыль, страховые компании должны собирать более высокие взносы на страхование, чем сумма, выплачиваемая застрахованному лицу. В связи с этим страховые компании вкладывают много времени, усилий и денег в создание прогнозирующих моделей, которые с высокой точностью могут предсказать расходы на здравоохранение каждого отдельного пациента.

В рамках научной статьи будет проведен анализ данных о страховых выплатах и построена модель прогнозирования размера этих

выплат. В качестве тестовых данных был взят набор данных с сайта Kaggle, где он распространяется в открытом доступе.

1. Задача регрессии

Регрессия [1] – зависимость математического ожидания (например, среднего значения) случайной величины от одной или нескольких других случайных величин (свободных переменных)

Регрессионный анализ – метод моделирования измеряемых данных и исследования их свойств. Данные состоят из пар значений зависимой переменной (переменной отклика) и независимой переменной (объясняющей переменной). Регрессионная модель есть функция независимой переменной и параметров с добавленной случайной переменной.

2. Решающие деревья

Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах – атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

Случайный лес является композицией (ансамблем [2]) множества решающих деревьев, что позволяет снизить проблему переобучения и повысить точность в сравнении с одним деревом. Прогноз получается в результате агрегирования ответов множества деревьев. Тренировка деревьев происходит независимо друг от друга (на разных подмножествах), что не просто решает проблему построения одинаковых деревьев на одном и том же наборе данных, но и делает этот алгоритм весьма удобным для применения в системах распределенных вычислений.

В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по следующей схеме:

- Выбирается подвыборка обучающей выборки размера `samplesize` – по ней строится дерево (для каждого дерева – своя подвыборка);

- Для построения каждого расщепления в дереве просматриваем количество случайных признаков (для каждого нового расщепления – свои случайные признаки);

- Выбираем наилучшие признаки расщепление по нему (по заранее заданному критерию).

Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление.

3. Метрика качества

В качестве методов оценки точности регрессионной модели принято использовать среднюю абсолютную ошибку [3], которая вычисляет модуль отклонения прогноза от истинного ответа и усреднить это по всем объектам обучающей выборки.

Такой функционал немного сложнее, чем среднеквадратичная ошибка. Его тяжелее оптимизировать из-за того, что у модуля производная есть не везде, в нуле она отсутствует. Но при этом здесь считается модуль отклонения, а не квадрат, и поэтому штраф за сильное отклонение гораздо меньше. Такой метод гораздо более устойчив к выбросам.

У среднеквадратичной ошибки [3] есть одна модификация – коэффициент детерминации, которая позволяет интерпретировать свое значение.

Основная часть коэффициента детерминации, или коэффициента R квадрат, – это дробь, у которой в числителе стоит сумма квадратов отклонений прогнозов алгоритма от истинных ответов, то есть практически среднеквадратичная ошибка, только без усреднения, просто сумма. В знаменателе стоит сумма квадратов отклонений истинных ответов от среднего истинного ответа, по всей обучающей выборке.

4. Получение попарных соответствий

Для построения более точной регрессионной модели необходимо провести первоначальный анализ данных и выявить наиболее влияющие признаки на размер выплаты. На рис. 1 предоставлен фрагмент прогнозируемых данных.

Рассмотрим поподробнее все признаки:

- Age – возраст пациента;
- Sex – пол пациента;
- BMI – индекс массы тела;
- Children – количество детей у пациента;
- Smoker – наличие вредных привычек;
- Region – регион проживания пациента;
- Charges – размер выплаты.

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	center	1725.55230
2	28	male	33.000	3	no	center	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

Рис. 1. Выборка из 5 объектов в наборе данных

Стоимость медицинских выплат зависит в разной степени от всех признаков, и чтобы получить наиболее точный результат в прогнозировании постараемся найти зависимость стоимости от каждого признака отдельно. Визуализация результатов анализа данных представлена на рис. 2-4.

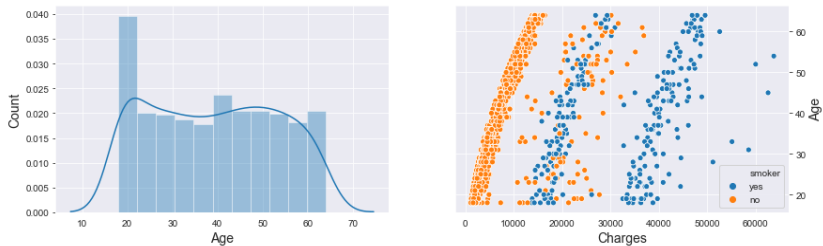


Рис. 2. Зависимость размеры платы от возраста пациент

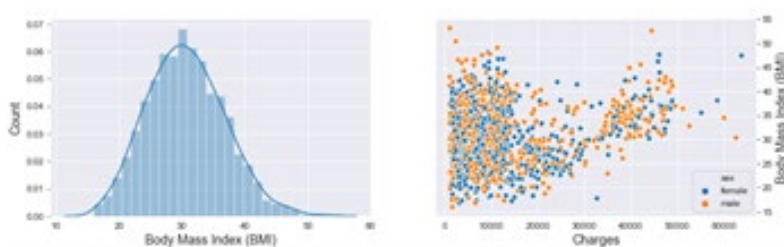


Рис. 3. Зависимость размера выплаты от индекса массы пациента

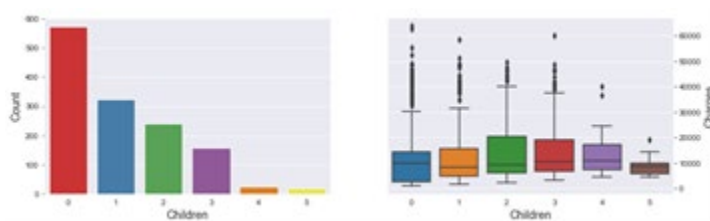


Рис. 4. Зависимость размера выплаты от количества детей у пациента

Исходя из проведенного анализа можно выделить 3 признака, имеющие наибольшее влияние на размер выплаты, а именно количество детей, возраст и индекс массы.

5. Результат построения модели

Обученная, методом случайного леса, модель прогнозирует с высокой точностью. Результаты построения модели представлены на рис. 5. Если разбираться более детально, это означает, что при сравнении реальных данных с прогнозируемыми значения практически совпадают.

Также, хотелось бы отметить значения среднеквадратичного и абсолютного отклонения. Числа получились весьма приближенные друг к другу, что может подтвердить точность прогноза.

```
CV: 0.8587777340167285
R2_score (train): 0.8909739985976602
R2_score (test): 0.8966862506108729
RMSE: 3948.3071552853057
```

Рис. 5. Метрика качества построенной прогнозирующей модели

Заключение

По результатам выполненной работы можно сделать следующие выводы:

- Нейронные сети являются очень мощным инструментом для работы на финансовых рынках, но для освоения технологии нужно потратить времени уже точно не меньше, чем на освоение технического анализа.

- Основные проблемы, возникающие при работе с этой технологией – правильная предобработка данных, этот этап играет решающую роль для прогнозирования данных.

В рамках данной работы был проведен детальный анализ данных, в ходе которого были выявлены самые значимые признаки, влияющие на размер стоимости выплаты.

С учетом этих признаков, была построена модель прогнозирования, выходные данные которой являются достаточно точными.

Список литературы

1. Воронцов, К. В. Лекции по машинному обучению / К. В. Воронцов [Электронный ресурс]. – Режим доступа : http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_%28курс_лекций%2C_К.В.Воронцов%29
2. Дьяконов, А. Г. Анализ малых данных. / А. Г. Дьяконов [Электронный ресурс]. – Режим доступа : <https://dyakonov.org/2019/04/19/ансамбли-в-машинном-обучении/>
3. Дьяконов, А. Г. Решение задач анализа данных / А. Г. Дьяконов [Электронный ресурс]. – Режим доступа : <http://jmllda.org/papers/doc/2013/JMLDA2013no5.pdf>

Алгоритмы смешивания клипов для трёхмерной скелетной анимации

Н. В. Пауков

Студент бакалавриата

Д. И. Соломатин

Старший преподаватель

Введение

Различные методы анимации персонажей и объектов окружения давно стали неотъемлемой частью компьютерных визуализаций, в особенности тех, которые выполняются в реальном времени.

В настоящее время существует несколько популярных технологий анимации. В контексте трёхмерной графики наиболее распространена так называемая скелетная анимация, которая стала стандартом индустрии в силу таких особенностей, как относительная простота реализации, высокая эффективность при рисовании с видеокартами, возможность использовать одни и те же анимационные клипы для разных моделей, а также существование большого количества инструментов для работы дизайнеров.

Однако, несмотря на перечисленные преимущества, использование скелетной анимации приводит к появлению некоторых трудностей. Так, данная технология плохо подходит для анимации тел, подверженных

упругим деформациям, а также для анимации с высокой степенью детализации (например, для лицевой анимации). Но наиболее существенным недостатком классической реализации, когда в каждый момент времени проигрывается единственный клип, является частая необходимость готовить большое количество избыточных вспомогательных клипов на основе уже существующих. Например, если для персонажа компьютерной игры созданы клипы ходьбы и прыжка, то также будет необходимо подготовить клипы перехода между этими состояниями. Если переход может происходить в любой момент времени, то для достижения его плавности может понадобиться создавать отдельные клипы для каждого кадра исходного состояния.

Целью данной работы является рассмотрение причин появления такой избыточности и обзор методов решения обозначенной проблемы путём комбинирования нескольких анимационных клипов с целью получить некоторое промежуточное состояние.

1. Причины появления избыточных клипов

Под избыточными клипами в данном тексте понимаются такие клипы, которые могли бы быть получены без непосредственного создания дизайнером с помощью применения различных алгоритмов. Однако перед рассмотрением данных алгоритмов необходимо более подробно рассмотреть ситуации, в которых имеет место быть упомянутая избыточность.

Во-первых, проблемы возникают в случаях, когда разрабатываются анимации, зависящие от направления, например, анимация взгляда в сторону для персонажа компьютерной игры. При классическом подходе с проигрыванием только одного клипа в каждый момент времени пришлось бы создать по одному клипу на каждое из возможных направлений.

Во-вторых, возможна ситуация, когда существуют несколько анимационных состояний, таких как бездействие, бег, медленная ходьба, и каждое из которых представлено отдельным клипом. Трудности здесь возникают при выполнении переходов от одного состояния к другому. При классическом подходе такие переходы будут являться резкими и приведут к «рывкам». Для их сглаживания дизайнеру необходимо создавать отдельные клипы для смены состояний, но ситуацию осложняет то, что, к примеру, переход между анимацией бездействия к анимации бега может произойти в любой момент времени при появлении внешнего события (при нажатии на клавишу или при движении мыши).

И, наконец, с обозначенными трудностями приходится сталкиваться в случае, когда состояния анимации могут быть

комбинированными (например, бег с предметом в руке и без или стояние на месте с предметом и без). В случае, когда таких комбинаций много, количество необходимых клипов может расти экспоненциально, что приводит к значительному увеличению сложности и времени реализации проекта.

При этом можно заметить, что подобные вспомогательные клипы обычно не являются самостоятельными и полностью или по большей части определяются теми клипами, на основе которых создаются. Так, например, если были смоделированы клипы взгляда вправо и влево, можно заметить, что все промежуточные направления взгляда можно получить как их комбинацию в разных пропорциях, тем самым освободив дизайнера от лишней работы. Механизм получения нового клипа на основе комбинации двух или более существующих называется смешиванием клипов и часто применяется при трёхмерной визуализации с применением скелетной анимации.

Существует несколько различных способов смешивания, таких как смешивание на основе линейной интерполяции, аддитивное смешивание, частичная замена ключевых кадров для костей одного клипа кадрами другого.

2. Смешивание на основе линейной интерполяции

Наиболее простым и универсальным алгоритмом смешивания является смешивание на основе линейной интерполяции, которая применяется к двум или, гораздо реже, трём и более клипам. В данном тексте будут рассматриваться только алгоритмы смешивания двух клипов, так как они естественным образом обобщаются на случаи большего количества.

Клип в данном случае представляет собой набор ключевых кадров для каждой из костей скелета модели (кость – материальная точка, которая является центром соответствующей системы координат). В общем случае оба клипа могут иметь разную длину и частоту кадров, поэтому также могут иметь и собственный локальный счётчик времени, который со временем увеличивается при отрисовке [1]. Целевой (итоговый) клип можно получить путём интерполяции в каждый момент времени поз скелета, соответствующих исходным клипам. Основным параметром такого смешивания является коэффициент интерполяции $0 \leq \alpha \leq 1$, который характеризует степень влияния первой позы на результат.

Существуют разные способы хранения поз в памяти компьютера, однако одним из наиболее практичных является представление позы как набора SRT (scale, rotation, translation) или RT (rotation, translation)

структур. Масштаб и перенос обычно представляют трёхмерный вектор, а поворот задаётся кватернионом, при этом данные величины заданы относительно родительской системы координат для кости (родительская кость или пространство модели для корневой кости). Математически, можно определить позу скелета для клипа A в момент времени t как:

$$P_t^A = \{(R_t^1, T_t^1), K, (R_t^n, T_t^n)\}, \quad (1)$$

где R – кватернион, задающий поворот, T – вектор позиции, n – количество костей скелета.

При интерполяции двух поз A и B с коэффициентом α итоговую позу можно получить, используя следующую формулу:

$$P_t' = \{(\text{slerp}(R_t^i, R_t^j, \alpha), \text{lerp}(T_t^i, T_t^j, \alpha))\}, \quad (2)$$

где slerp – функция сферической линейной интерполяции кватернионов, а lerp – функция стандартной линейной интерполяции векторов. Данные функции определяются следующим образом:

$$\text{slerp}(q_1, q_2, \alpha) = \frac{\sin(\Omega(1-\alpha))}{\sin(\Omega)} q_1 + \frac{\sin(\Omega\alpha)}{\sin(\Omega)} q_2, \quad (3)$$

здесь $\cos(\Omega) = q_1 q_2$.

Сферическая интерполяция требует больше вычислений, чем обычная линейная, но при этом позволяет достичь лучшего визуального результата [2].

Линейная интерполяция векторов аналогична линейной интерполяции скаляров:

$$\text{lerp}(v_1, v_2, \alpha) = (1-\alpha)v_1 + \alpha v_2, \quad (4)$$

На рис. 1 продемонстрирована схема использования описанного метода на примере скелета, состоящего из трёх костей и двух поз (слева и справа). В результате интерполяции с коэффициентом 0.5 получается промежуточное среднее положение.

Стоит отметить, что данный метод смешивания может использоваться не только для непосредственной интерполяции различных клипов для получения нового анимационного состояния. Его также можно использовать для реализации плавных переходов между двумя различными состояниями.

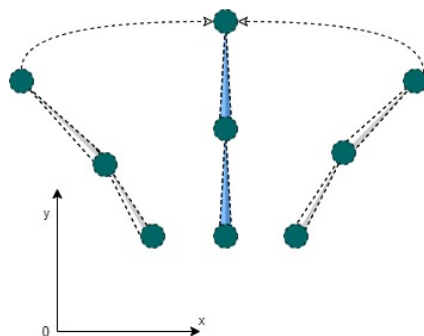


Рис. 1. Схема применения линейной интерполяции поз

Для этого при начале перехода можно запомнить последнюю позу в старом состоянии, начать проигрывать клип, соответствующий новому состоянию, но при отрисовке в течение некоторого короткого промежутка времени линейно интерполировать сохранённую и текущую позы с коэффициентом, увеличивающимся до единицы с течением времени. В таком случае получится избежать резкого «рывка» изображения. Другим способом реализовать плавный переход является начало воспроизведения нового клипа перед остановкой воспроизведения старого и интерполяция двух полученных поз с коэффициентом, значение которого также будет возрастать с течением времени, пока не достигнет единицы.

3. Аддитивное смешивание

Аддитивное смешивание представляет собой алгоритм, использующий линейную интерполяцию, описанную выше, но делающий это иначе.

В соответствии со своим названием, данный способ заключается в прибавлении некоторого добавочного клипа к исходному клипу с коэффициентом влияния $0 \leq \alpha \leq 1$.

Добавочный клип представляет собой такой клип, в котором SRT- или RT-структуры костей вместо информации о непосредственном положении хранят информацию об относительном изменении своих параметров, необходимом для перехода из исходного клипа к целевому. Многие 3D-редакторы не позволяют напрямую создавать такие клипы, однако, при реализации приложения, поддерживающего данный способ анимации, можно получить добавочный клип путём вычитания исходного клипа из целевого.

Основная сложность здесь заключается в том, что стандартные операции вычитания и умножения не определены для RT-структур, а для матриц приведут к некорректным результатам при выполнении линейной интерполяции. Для замены в случае матричного представления можно использовать умножение и обращение матриц, а в случае RT-структур – операции с отдельными компонентами.

При проигрывании анимации в момент времени t текущая поза в результате вычислений по формуле (2) линейной интерполяции двух поз с единственным отличием – в качестве второй позы B используется поза, полученная в результате комбинации исходного и добавочного клипов.

Основным преимуществом данного метода перед стандартной линейной интерполяцией является возможность прибавлять добавочный клип, полученный из одного исходного клипа к другим исходным клипам. Например, если добавочный клип был получен путём вычитания из клипа ходьбы с размахиванием рукой клипа ходьбы с опущенной рукой, то полученный результат можно будет прибавить к клипу бега и получить, таким образом, клип бега с размахиванием рукой [3].

4. Замена части анимации

Следующим методом, используемым для смешивания анимационных клипов, является полная замена части одного клипа частью другого. Так, например, если уже существуют клип ходьбы с опущенной рукой и клип размахивания рукой, то можно получить клип ходьбы с размахиванием рукой путём выборки части ключевых кадров из первого клипа и части из второго. Параметром данного вида смешивания является битовая маска, определяющая источник ключевых кадров для костей скелета.

$$M = (v_1, K, v_n), \quad (5)$$

где v_i равняется нулю, если источником служит первый клип и единице, если второй.

Целевая поза кости скелета с индексом i будет определяться так:

$$B'_{i,t} = \begin{cases} P_{i,t}^A, M_i = 0 \\ P_{i,t}^B, M_i \neq 0 \end{cases}, \quad (6)$$

Данный метод отличается простотой реализации, но при этом может приводить к нежелательным эффектам несвязности одной части анимации с другой. Для компенсации подобных недостатков можно использовать не полную замену анимации отдельных костей, а

частичную, например, используя вместо битовой маски вектор коэффициентов влияния и выполняя линейную интерполяцию поз разных костей с различными весами.

5. Деревья смешивания

Данный метод не является самостоятельным методом смешивания клипов, но позволяет объединять описанные выше методы в последовательность преобразований, чтобы добиться наилучшего результата.

Дерево смешивания представляет собой такое дерево, листья которого являются анимационными клипами, а промежуточные вершины – различными операциями смешивания и преобразования поз. Такой механизм позволяет создавать сложные анимационные состояния, используя относительно небольшое число созданных дизайнером клипов. Пример подобного дерева продемонстрирован на рис. 2.

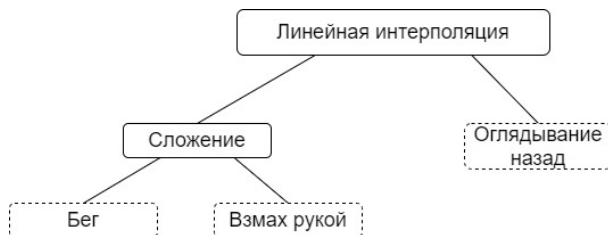


Рис. 2. Дерево смешивания

6. Особенности реализации

Описанные подходы к смешиванию обладают своими особенностями, но основываются на общей части - линейной интерполяции поз скелета или отдельной кости. В листинге ниже продемонстрирована реализация обоих указанных действий.

Листинг

Линейная интерполяция поз

```
BonePose interpolate(BonePose first, BonePose second, float factor)
{
    BonePose result;
    result.position = lerp(first.position, second.position, factor);
    result.orientation = slerp(first.orientation, second.orientation, factor);

    return result;
}
```

```

AnimationPose interpolate(AnimationPose first, AnimationPose
second, float factor)
{
    AnimationPose result(first.getSkeleton());
    for (uint8_t index = 0; index < first.getBonesCount();
index++) {
        auto targetPose = interpolate(first.getBonePose(index),
second.getBonePose(index), factor);
        result.setBonePose(index, targetPose);
    }
}

```

На рис. 3 продемонстрировано применение линейной интерполяции к двум позам, соответствующим клипам взгляда влево и вправо.

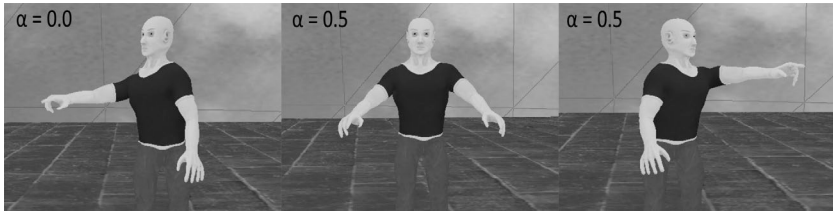


Рис. 3. Пример применения линейной интерполяции поз

Заключение

Таким образом, в данном тексте было предложено решение проблемы создания большого количества избыточных клипов при использовании скелетной анимации с помощью смешивания. Были рассмотрены основные используемые на практике алгоритмы смешивания, а также продемонстрирована их реализация на языке программирования C++.

Список литературы

1. Seron, F. J. Adding support for high-level skeletal animation / F. J. Seron // IEEE Transactions on Visualization and Computer Graphics. – 2002. – № 8. – P. 360-372.
2. Marucchi-Foino, R. Game and Graphics Programming for IOS and Android with OpenGL ES 2.0 / R. Marucchi-Foino. – В. : Wrox, 2012. – 308 p.
3. Gregory, J. Game engine architecture / J. Gregory. – 3-rd ed. – L. : CRC Press, 2018. – 1240 p.

Формирование предварительного коммерческого предложения ООО «ЧерноземАгромаш» средствами MS Excel

К. Д. Подольский

Студент бакалавриата

Я. Э. Головин

Студент бакалавриата

Е. А. Копытина

Ассистент

Введение

Коммерческое предложение – один из самых мощных инструментов продаж. Грамотно составленное, красиво оформленное, структурированное и содержащее только самую необходимую информацию коммерческое предложение лучше всего влияет на клиента. Но, к сожалению, менеджеры не уделяют его составлению должного внимания, а тратят большое количество денежных средств только на рекламу. Программа для подготовки предварительного коммерческого предложения для автоматизации работы менеджера ООО «ЧерноземАгромаш» средствами MS Excel будет рассмотрена в данной статье.

1. Постановка задачи

Имеются документы в электронном виде, с которыми ежедневно работает менеджер ООО «ЧерноземАгромаш», копируя и вставляя данные из одного документа в другой, что, естественно, приводит к ошибкам в расчетах и медленному формированию предварительного коммерческого предложения. Необходимо разработать приложение, которое будет работать с двумя электронными таблицами, расположенными в разных документах. Требуемые функциональные возможности следующие:

- Возможность выбора наименования оборудования в документе формирования коммерческого предложения и автоматическая вставка соответствующей цены, которые берутся из документа с перечнем всего имеющегося оборудования.

- Вставка текущей даты и курса евро с сайта ЦБ РФ.

2. Реализация логики

Имеется два документа:

- Оборудование.xls.
- КР.xls.

Оборудование.xls содержит в себе всю информацию об оборудовании, которое продает ООО «ЧерноземАгромаш», а именно: наименование, единицы измерения, имеющееся на складе количество и цена за единицу. Таким образом, Оборудование.xls является неким подобием базы данных.

КР.xls представляет из себя отформатированный структурированный шаблон предварительного коммерческого предложения.

Реализация логики выполнялась на языке VBA. Таким образом, управление электронными таблицами осуществляется посредством объектно-ориентированной модели кода и данных. С помощью этого кода данные входных таблиц будут мгновенно обрабатываться и отображаться в таблицах. Таблица становится интерфейсом кода, позволяя легко работать, изменять его и управлять расчётами [1].

В документе КР.xls были реализованы следующие процедуры:

- Function GetRate(...).
- Vd(...)

В документ КР.xls с помощью процедуры Vd() через сформированную форму выбора позиций переносится необходимая информация по выбранному менеджером наименованию оборудования и соответствующая цена из документа Оборудование.xls, а Function GetRate(...) вставляет текущую дату и курс евро в документ КР.xls с сайта ЦБ РФ с помощью протокола HTTP, как аналогичным образом реализовано в [2, 3].

Ниже приводится основная часть программного кода процедуры Function GetRate(...) (листинг).

Листинг

Программный код функции Function GetRate(...)

```
Function GetRate(ByVal CurrencyName As String, ByVal RateDate
As Date) As Double
    On Error Resume Next
    CurrencyName = UCase(CurrencyName): If Len(CurrencyName)
<> 3 Then Exit Function
    Set xmldoc = CreateObject("Msxml.DOMDocument"):
xmldoc.async = False
    url_request =
"http://www.cbr.ru/scripts/XML_daily.asp?date_req=" +
Format(RateDate, "dd\mm\yyyy")
```

Архитектуру разработанной программы можно представить с помощью схемы, которая показана на рис. 1.

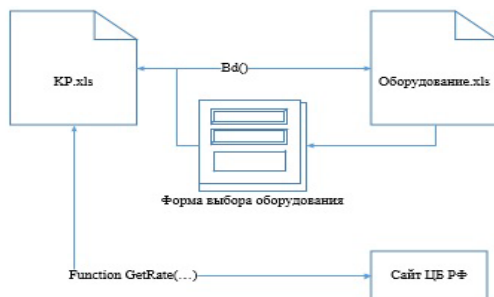


Рис. 1. Схема логики работы программы

3. Реализация интерфейса

При открытии структурированного шаблона коммерческого предложения KR.xls автоматически проставляется дата и текущий курс евро с сайта ЦБ РФ (рис. 2).

ООО «ЧерноземАгромаш»
394030, г. Воронеж,
ул. Плеванская, 53, 14 этаж
+7 (473) 269-42-35
http://chernozemagromash.ru
info@agrovni.com

Строим с Уважением к Вам и Вашему бизнесу

Предварительное коммерческое предложение
Зернохранилище вместимостью

Заказчик:
Адрес объекта:
Исполнитель: ООО «ЧерноземАгромаш»
г. Воронеж Курс евро 79.1189 руб. 06.05.2020

Узел приема зерна
Спецификация ответственного оборудования

№	Наименования оборудования	Ед. изм.	Кол.	Цена за ед. изм., руб.	Сумма, руб.
1		шт.	1		0,00р.
2		шт.	1		0,00р.
3		компл.	1		0,00р.
4		шт.	1		0,00р.
6		шт.	1		0,00р.
7		мЗ	1		0,00р.
Итого ответственного оборудования по узлу:					0,00р.
Итого ответственного оборудования:					0,00р.

Спецификация импортного оборудования

Рис. 2. Дата и курс евро с сайта ЦБ РФ

Затем менеджер начинает вводить искомое наименование оборудования и запускает макрос при нажатии на кнопку на панели быстрого доступа (рис. 3).

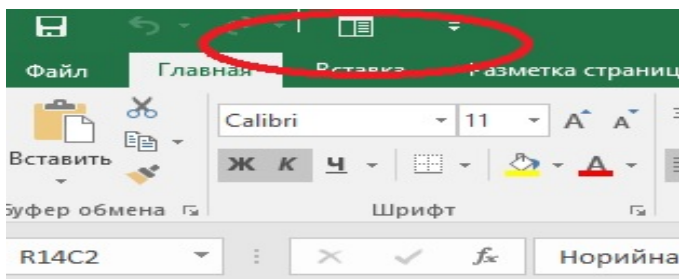


Рис. 3. Запуск макроса кнопкой на панели быстрого доступа Менеджер видит перед собой форму выбора значения (рис. 4).

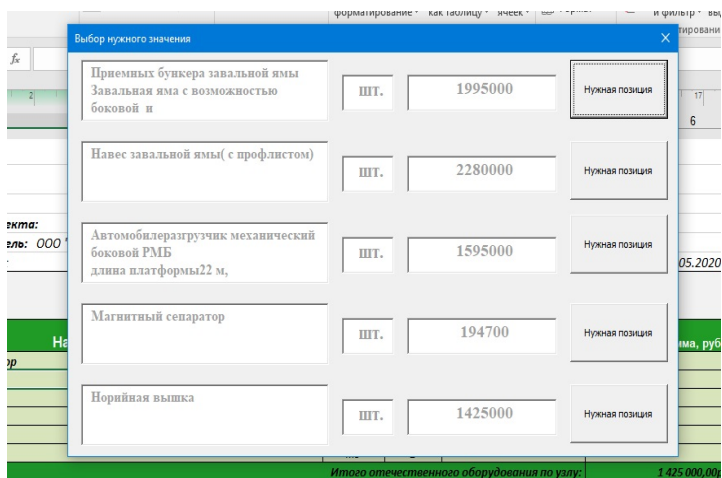


Рис. 4. Форма выбора подобранного оборудования

При нажатии на кнопку формы выбора «Нужная позиция» полное наименование оборудования, единицы измерения и цена переносятся в документ KP.xls из документа Оборудование.xls (рис. 5).

Заключение

В данной статье рассматривается реализация программы для формирования предварительного коммерческого предложения, которая написана на языке VBA и обладает следующими функциональными возможностями:

- Производит вставку текущего курса евро и даты с сайта ЦБ РФ.

- Позволяет перенести данные о введенном наименовании оборудования, которое берется из документа с перечнем всего имеющегося у организации оборудования.

Благодаря написанному макросу, менеджер ООО «ЧерноземАгромаш» сократил в несколько раз не только время формирования предварительного коммерческого предложения и ошибки, которые возникали в его работе ранее по причине постоянного переноса данных.

Предварительное коммерческое предложение						
Зернохранилище емкостью						
Заказчик:						
Адрес объекта:						
Исполнитель: ООО "ЧерноземАгромаш"						
г. Воронеж		Курс евро: 80,0598		на		07.05.2020
Узел приема зерна						
Спецификация отечественного оборудования						
№ п/п	Наименование оборудования	Ед. изм.	Кол-во	Цена за ед. изм., руб.	Сумма, руб.	
1	Норийная вышка	шт.	1	1 425 000,00р.	1 425 000,00р.	
2		шт.	1		0,00р.	
3		компл.	1		0,00р.	
4		шт.	1		0,00р.	
6		шт.	1		0,00р.	
7		м3	1		0,00р.	
Итого отечественного оборудования по узлу:					1 425 000,00р.	
в том числе НДС:					217 372,88р.	

Рис. 5. Вставка данных о выбранном оборудовании

Список литературы

1. Копытина, Е. А. Оптимизация стоимости доставки ресурсов при строительстве инженерных коммуникаций / Е. А. Копытина, Н. А. Петрикеева // ВМ. Проектирование. Строительство. Эксплуатация Материалы Всероссийского форума; под редакцией Д. К. Проскурина. – 2018. – С. 51-55.
2. Петрикеева, Н. А. Вариант решения задачи оптимизации толщины теплоизоляционного слоя в теплоснабжении / Н. А. Петрикеева, Д.М. Чудинов, Е.А. Копытина, Л.П. Мышовская // Научный журнал строительства и архитектуры. – 2018. – № 2 (50). – С. 21-28.
3. Куликова, Е. А. Оптимизация работы энергосистем. Определение вероятного ущерба от перерывов энергоснабжения / Е. А. Куликова, Н. М. Попова, Н. В. Коротких, Н. А. Петрикеева // Градостроительство. Инфраструктура. Коммуникации. – 2018. – № 4 (13). – С. 29-36.

Сравнительный анализ различных способов реализации метода канонических корреляций для цифровых сигналов

В. Н. Попов

Студент бакалавриата

Е. А. Киселев

Доцент

Введение

В настоящее время метод канонических корреляций имеет достаточно широкое применение для обработки статистических данных различной природы [1]. Его особенностью является то, что он позволяет находить максимальные взаимосвязи между двумя наборами случайных величин или экспериментальных сигналов. На практике при работе с большими объемами информации метод требует привлечения серьезных вычислительных ресурсов, поэтому поиск эффективных алгоритмов его программной реализации является актуальной задачей.

В данной работе будут рассмотрены два способа расчета канонических корреляций, адаптированных для работы с цифровыми сигналами. Первый требует нахождения собственных значений вспомогательной матрицы, которая строится из блоков матрицы ковариации двух исследуемых наборов сигналов. При этом на практике необходимо производить расчет двух обратных матриц большого размера [1]. Второй способ был предложен нами в статье [2]. Он основан на применении процедуры ортонормализации Грама–Шмидта. В рамках данной работы будет проведен сравнительный анализ вычислительной сложности этих двух способов в различных ситуациях, с целью выявить области их наиболее эффективного применения.

1. Основные определения и обозначения

Пусть X – некоторая случайная величина, ее математическое ожидание мы будем обозначать стандартным образом: $M[X]$. Случайные величины можно рассматривать как элементы гильбертова пространства со скалярным произведением

$$(X, Y) = M[XY]. \quad (1)$$

Для удобства условимся, что в дальнейшем мы будем работать с предварительно центрированными, $M[X] = 0$, и нормированными,

$\|X\| = \sqrt{(X, X)} = 1$, величинами. Тогда скалярное произведение X и Y будет равно их коэффициенту корреляции.

Рассмотрим два набора случайных величин $X_k, k = 1, 2, \dots, m$, и $Y_k, k = 1, 2, \dots, n$. В методе канонических корреляций требуется найти такие их нормированные линейные комбинации U_1 и V_1 , коэффициент корреляции между которыми будет наибольшим. Таким образом, необходимо построить две величины

$$U_1 = \sum_{k=1}^m \alpha_k^{(1)} X_k, \quad V_1 = \sum_{k=1}^n \beta_k^{(1)} Y_k, \quad (2)$$

где $\alpha_k^{(1)}, \beta_k^{(1)}$ – неопределенные коэффициенты, удовлетворяющие следующим условиям:

$$\begin{cases} (U_1, V_1) = \max, \\ (U_1, U_1) = 1, (V_1, V_1) = 1. \end{cases} \quad (3)$$

U_1 и V_1 называются первой парой канонических случайных величин. Далее вводится понятие второй пары и т. д. В рамках данной работы мы ограничимся рассмотрением только процедуры нахождения U_1 и V_1 .

Пусть A_1 – вектор-столбец, составленный из коэффициентов $\alpha_k^{(1)}$, а B_1 – из $\beta_k^{(1)}$. Матрицу ковариации всего набора случайных величин X_k и Y_k обозначим R . Ее удобно разбить на блоки следующим образом:

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}, \quad (4)$$

где $(R_{11})_{ij} = (X_i, X_j)$, $(R_{22})_{ij} = (Y_i, Y_j)$, $(R_{12})_{ij} = (X_i, Y_j)$, $R_{21} = R_{12}^T$.

2. Способы расчета канонических корреляций

Первый способ, применяемый в настоящее время наиболее широко, состоит из следующих этапов [1].

1. Построение вспомогательной матрицы

$$C = R_{11}^{-1} R_{12} R_{22}^{-1} R_{21}. \quad (5)$$

2. Расчет собственных значений C . Наибольшее из них λ_{\max} равно квадрату максимального коэффициента корреляции.

3. Нахождение канонических величин. Величина A_1 будет нормированным собственным вектором матрицы C , соответствующим

собственному значению λ_{\max} . Вектор-столбец B_1 можно найти по формуле

$$B_1 = \frac{1}{\sqrt{\lambda_{\max}}} R_{22}^{-1} R_{21} A_1. \quad (6)$$

Второй способ, который предложен в статье [2], заключается в следующем.

1. Ортонормализация методом Грама–Шмидта отдельно в пределах каждого набора X_k и Y_k . Полученные случайные величины мы обозначим X_k° и Y_k° , а соответствующую им матрицу ковариации – R° . Канонические величины тогда удобно записать в виде линейной комбинации X_k° и Y_k° :

$$U_1 = \sum_{k=1}^m \alpha_k^{(1)} X_k^{\circ}, \quad V_1 = \sum_{k=1}^n \beta_k^{(1)} Y_k^{\circ}. \quad (7)$$

4. Расчет собственных значений матрицы $C^{\circ} = R_{12}^{\circ} R_{21}^{\circ}$. Наибольшее из них λ_{\max} равно квадрату максимального коэффициента корреляции.

5. Нахождение канонических величин. В данном подходе вектор-столбец X_1° , составленный из коэффициентов $\alpha_k^{(1)}$, будет нормированным собственным вектором матрицы C° , отвечающим λ_{\max} . Вектор-столбец B_1° , составленный из $\beta_k^{(1)}$, тогда равен

$$B_1^{\circ} = \frac{1}{\sqrt{\lambda_{\max}}} R_{21}^{\circ} X_1^{\circ}. \quad (8)$$

Отметим, что есть и множество других способов расчета канонических величин. Один из них основан на методе множителей Лагранжа [1]. Этот способ мы исключили из рассмотрения, поскольку он приводит к необходимости решения уравнения степени $m+n$, что является достаточно трудной задачей. Еще один подход, предложенный в работе [3], основан на построении вспомогательной матрицы, которая вычисляется аналогично формуле (5), но при этом содержит матричные сомножители степени $1/2$. Нахождение квадратного корня из матрицы требует приведения ее к диагональному виду, из-за чего вычислительная сложность алгоритма значительно возрастает [4]. Поэтому данный подход мы также не будем рассматривать в качестве основного.

3. Сравнительный анализ двух способов расчета первой пары канонических величин

Рассмотрим в качестве случайных величин некоторые цифровые сигналы, состоящие из N значений. Различия двух исследуемых способов расчета проявляются в основном на этапе 1. Подсчитаем оценочно количество операций умножения и сложения в рамках обоих подходов на этом этапе, при этом для простоты положим $m = n$.

Для реализации первого способа на этапе 1 необходимо сначала вычислить элементы матрицы ковариации R . Учитывая симметричность R , это потребует расчета $2m(2m-1)/2 \sim 2m^2$ скалярных произведений сигналов, что равносильно выполнению $\sim 2m^2N$ умножений и $\sim 2m^2N$ сложений. Если использовать метод Гаусса, то для нахождения обратных матриц R_{11}^{-1} и R_{22}^{-1} в совокупности необходимо $\sim 2m^3$ операций обоих видов и $\sim 3m^3$ для вычисления произведения $R_{11}^{-1}R_{12}R_{22}^{-1}R_{21}$. В результате получаем примерную оценку количества операций $\sim 2m^2N + 5m^3$ по числу умножений и столько же по числу сложений.

При использовании второго способа на этапе 1 необходимо предварительно провести ортонормализацию X_k и Y_k . Если применить метод Грама–Шмидта, то в совокупности для двух наборов сигналов нужно вычислить $\sim m^2$ скалярных произведений, что равносильно выполнению $\sim m^2N$ сложений и умножений. При поэлементном вычитании и умножении сигналов на коэффициенты потребуется еще $\sim m^2N$ операций обоих видов. Далее необходимо $\sim m^2N$ умножений и сложений при вычислении R_{12}' и $\sim m^3$ для нахождения $C_k' = R_{12}'R_{21}'$. В результате имеем $\sim 3m^2N + m^3$ умножений и столько же сложений.

Сравнивая обе полученные оценки, видим, что второй способ при достаточно больших значениях m выгоднее по числу операций умножения и сложения, если $N < 4m$, т. е. когда величины m и N примерно одного порядка. При больших значениях N эффективнее будет работать первый способ.

Рассмотрим теперь этап 3, на котором производится расчет первой пары канонических величин. Во втором подходе величины U_1 и V_1 по формулам (7) записаны в виде линейной комбинации X_k' и Y_k' . Если требуется представить U_1 и V_1 в виде суперпозиции исходных величин

X_k и Y_k , это будет неудобно, и предпочтение в таком случае лучше отдать первому способу.

Заключение

В данной работе проведен сравнительный анализ двух способов расчета канонических корреляций для цифровых сигналов. В качестве основных параметров рассматривались количество сигналов m и длина N каждого из них. Как показал теоретический анализ вычислительной сложности, при нахождении максимального коэффициента корреляции первый способ, который основан на обращении матриц, выгоднее использовать в случаях, когда N существенно превосходит m . Если величины N и m примерно одного порядка, то меньшее число арифметических операций содержит второй способ, основанный на ортогонализации. Также при нахождении канонических величин существенную роль играет постановка задачи. Если их требуется выразить через исходный набор сигналов, то удобнее первый способ. Если объектом интереса являются лишь сами канонические величины, то при $N < 4m$ выгоднее применить второй способ.

В дальнейшем планируется использовать результаты данной работы для создания программных средств, которые автоматически смогут выбирать оптимальный алгоритм расчета канонических корреляций в каждой конкретной ситуации, с целью повышения быстродействия при работе с большими объемами данных.

Список литературы

1. Дубров, А. М. Многомерные статистические методы / А. М. Дубров, В. С. Мхитрян, Л. И. Трошин. – М. : Финансы и статистика, 2003. – 352 с.
2. Максимов, А. В. Расчет канонических корреляций с помощью процедуры ортогонализации / А. В. Максимов, В. Н. Попов, Е. А. Киселев // ИНФОС-2019: матер. десятой междунар. науч.-техн. конф. (Вологда, 28-29 июня 2019 г.). – Вологда, 2019. – С. 41-44.
3. Борзых, Д. А. О методе канонических корреляций : препринт WP2/2016/02 / Д. А. Борзых. – М. : Изд. дом Высшей школы экономики, 2016. – 18 с.
4. Ильин, В. А. Линейная алгебра / В. А. Ильин, Э. Г. Позняк. – М. : Наука, 1984. – 296 с.

Построение ортогональных систем целочисленных сдвигов

В. О. Саввина

Студент бакалавриата

Е. А. Киселев

Доцент

Введение

В настоящее время разложение в ряд по различным ортогональным семействам функций находит весьма широкое применение. На этом основан анализ Фурье [1], теория вейвлетов [2] и другие важные прикладные разделы математики.

На данный момент существует несколько основных методов получения ортогональных систем: нахождение собственных функций некоторого самосопряженного оператора (например, оператора Штурма–Лиувилля [3]), решение масштабирующего уравнения в теории вейвлетов [2], проведение ортогонализации определенного набора функций (например, методом Грама–Шмидта [4]). Достоинством первого подхода является то, что набор собственных функций самосопряженного оператора образует базис в рассматриваемом гильбертовом пространстве, а недостатком то, что построенные семейства функций могут иметь достаточно сложную структуру. При использовании второго способа построения ортогональных систем также получается некоторый базис, но он изначально имеет вполне четкую структуру: состоит из сдвигов, растяжений и сжатий функций. В качестве недостатка вейвлетов можно отметить то, что они обычно строятся лишь в виде некоторых рядов, а простые явные формулы получить удается достаточно редко. Третий подход также в ряде случаев позволяет добиться определенной структуры ортогонализованного набора функций (это возможно, например, для систем равномерных сдвигов [2]), но он обычно приводит к формулам, которые записываются с помощью бесконечных рядов, а их суммирование, как правило, можно провести только приближенно.

В данной работе предложен новый способ построения ортогональных систем функций, имеющих структуру сдвигов. Подобный математический аппарат является достаточно удобным для области цифровой обработки сигналов, поскольку обычно измерения проводят с некоторым постоянным шагом. В общем случае получаемые

наборы функций не будут образовывать базис в пространстве $L_2(R)$, но на практике это не играет принципиальной роли: для дискретных сигналов в конечном счете все равно приходится использовать ограниченные наборы функций. В ряде ситуаций предлагаемый способ позволяет получать достаточно простые аналитические формулы.

1. Вывод основных соотношений

Мы будем рассматривать пространство $L_2(R)$. Скалярное произведение двух функций $f, g \in L_2(R)$ зададим обычным образом

$$(f, g) = \int_{-\infty}^{\infty} f(x) \overline{g(x)} dx, \quad (1)$$

где $\overline{g(x)}$ обозначает комплексное сопряжение. Требуется построить ортонормированное семейство сдвигов $\phi(\cdot - k)$, $k \in Z$, в котором $\phi \in L_2(R)$. В этом случае должно иметь место следующее соотношение:

$$(\phi(\cdot - m), \phi(\cdot - n)) = \int_{-\infty}^{\infty} \phi(x - m) \overline{\phi(x - n)} dx = \delta_{mn}, m, n \in Z, \quad (2)$$

где δ_{mn} – символ Кронекера. Переобозначив $t = x - m$, $k = n - m$, легко убедиться, что условие (2) эквивалентно равенству

$$(\phi, \phi(\cdot - k)) = \int_{-\infty}^{\infty} \phi(t) \overline{\phi(t - k)} dt = \delta_{0k}, k \in Z. \quad (3)$$

Заменим целое число k в формуле (3) непрерывной переменной y и рассмотрим функцию

$$\lambda(y) = \int_{-\infty}^{\infty} \phi(t) \overline{\phi(t - y)} dt. \quad (4)$$

Выражение (4) по структуре представляет из себя свертку $\phi(t)$ и $\overline{\phi(-t)}$. Зададим прямое и обратное преобразование Фурье следующим образом:

$$\hat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\xi t} dt, f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\xi) e^{i\xi t} d\xi. \quad (5)$$

Тогда формула (4) в образах Фурье примет следующий вид:

$$\sqrt{2\pi} \hat{\phi}(\xi) \overline{\hat{\phi}(\xi)} = \hat{\lambda}(\xi). \quad (6)$$

Отсюда можно выразить функцию $\hat{\phi}(\xi)$:

$$\hat{\phi}(\xi) = \frac{1}{\sqrt[4]{2\pi}} \sqrt{\hat{\lambda}(\xi)}. \quad (7)$$

Отметим, что в формуле (7) извлечь корень можно несколькими способами, в результате чего получатся различные $\hat{\phi}(\xi)$, отличающиеся фазовым множителем $e^{i\alpha(\xi)}$, где $\alpha(\xi)$ – некоторая вещественная функция, причем $\alpha(\xi + 2\pi) = \alpha(\xi)$. В дальнейшем в соотношении (7) мы будем использовать корень в смысле главного значения.

С помощью равенства (7), выбирая различные $\lambda(y)$, удовлетворяющие условию $\lambda(k) = \delta_{0k}$, можно находить функцию $\phi(t)$, сдвиги которой будут ортогональны.

2. Примеры построенных ортогональных систем

Рассмотрим несколько примеров. Пусть

$$\lambda(y) = (\text{sinc}(\pi y))^n = \left(\frac{\sin(\pi y)}{\pi y} \right)^n, n = 1, 2, \dots \quad (8)$$

Тогда, согласно формуле (7), имеем

$$\hat{\phi}(\xi) = \frac{1}{\sqrt{2\pi}} \sqrt{B_n \left(\frac{\xi}{2\pi} \right)}, \quad (9)$$

где $B_n(x)$ – центрированный B -сплайн порядка n . С помощью свертки $B_n(x)$ рекуррентно задаются следующим образом [2]:

$$B_1(x) = \chi_{[-0.5, 0.5]}(x), \quad (10)$$

$$B_{n+1}(x) = \int_{-\infty}^{\infty} B_n(x-t) B_1(t) dt, n = 1, 2, \dots, \quad (11)$$

где $\chi_{[a,b]}(x)$ – характеристическая функция отрезка $[a, b]$.

Если вычислить обратное преобразование Фурье функции (9), то при $n = 1$ получим

$$\phi(t) = \text{sinc}(\pi t). \quad (12)$$

В случае $n = 2$ получается уже менее тривиальный результат, выражающийся через гипергеометрическую функцию:

$$\phi(t) = \frac{4}{3} {}_1F_2 \left(1; \frac{5}{4}, \frac{7}{4}; -\pi^2 t^2 \right). \quad (13)$$

График функции (13) показан на рисунке. К сожалению, при $n > 2$ аналитически обратное преобразование Фурье функции (9) найти не удалось.

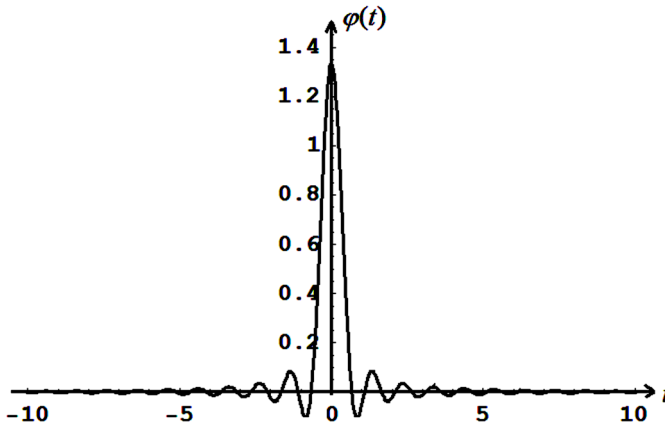


Рисунок. График функции $\varphi(t)$, определяемой формулой (13)

С помощью равенства (7) можно построить множество других примеров ортогональных систем сдвигов, если выбрать $\lambda(y)$ в виде

$$\lambda(y) = g(y)\text{sinc}(\pi y), \quad (14)$$

где $g \in L_2(\mathbb{R})$, $g(0) = 1$. Например, если $g(y) = e^{-|y|}$, то в образах Фурье придем к следующему равенству:

$$\hat{\phi}(\xi) = \frac{1}{\pi\sqrt{2}} \sqrt{\text{arctg}(\xi + \pi) - \text{arctg}(\xi - \pi)}. \quad (15)$$

Если выбрать $g(y) = \exp(-y^2)$, то получим

$$\hat{\phi}(\xi) = \frac{1}{2\sqrt{\pi}} \sqrt{\text{erf}\left(\frac{\xi + \pi}{2}\right) - \text{erf}\left(\frac{\xi - \pi}{2}\right)}, \quad (16)$$

где $\text{erf}(x)$ – функция ошибок (интеграл вероятностей) [5]:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt. \quad (17)$$

К сожалению, в обоих рассмотренных случаях найти аналитически обратное преобразование Фурье также не удалось.

Заключение

В данной работе предложен новый способ построения ортогональных систем целочисленных сдвигов. Его достоинством является то, что в образах Фурье он позволяет получать достаточно простые аналитические соотношения. Найти обратное преобразование Фурье и построить ортогональную систему в явном виде, к сожалению, в большинстве случаев не удается, но эту трудность на практике можно обойти. Для этого при вычислении скалярных произведений исследуемый сигнал необходимо предварительно подвергнуть преобразованию Фурье, а затем воспользоваться его унитарностью (равенством Парсеваля или теоремой Планшереля) [2]. Также среди недостатков разработанного подхода следует отметить то, что получаемые ортогональные наборы функций в общем случае не образуют базис в пространстве $L_2(R)$, что является важным аспектом при исследовании многих теоретических вопросов в рамках функционального анализа, квантовой физики и т. д. Однако в большинстве задач цифровой обработки сигналов отсутствие полноты не является принципиальным обстоятельством, поэтому построенные в данной работе системы функций вполне могут быть востребованы в этой области.

В дальнейшем планируется построить и другие примеры ортогональных семейств сдвигов, рассмотреть более общие оконные системы функций, а также применить разработанный математический аппарат для сжатия и анализа цифровых сигналов.

Список литературы

1. Ряды. Интеграл Фурье и преобразование Фурье. Приложения / И. М. Аксененкова [и др.]. – М. : Ленанд, 2020. – 240 с.
2. Чуи, Ч. Введение в вейвлеты / Ч. Чуи; пер. с англ. Я. М. Жилейкина. – М. : Мир, 2001. – 412 с.
3. Владимиров, В. С. Уравнения математической физики / В. С. Владимиров, В. В. Жаринов. – М. : ФИЗМАТЛИТ, 2004. – 400 с.
4. Кострикин, А. И. Линейная алгебра и геометрия / А. И. Кострикин, Ю. И. Манин. – М. : Наука, 1986. – 304 с.
5. Абрамовиц, М. Справочник по специальным функциям / М. Абрамовиц, И. Стиган ; пер. с англ. В. А. Диткина и Л. Н. Кармазиной. – М. : Наука, 1979. – 832 с.

Разработка программного модуля для решения задачи сегментации полупрозрачных объектов с потерей фокусировки на текстовых изображениях с его последующим восстановлением

Д. Е. Семенихина

Студент бакалавриата

А. Ю. Иванков

Ассистент

Введение

В настоящее время непрерывно совершенствуются технологии в области получения изображений, а также значительно повышается вычислительный потенциал современных компьютеров, что обуславливает появление новых методов и алгоритмов автоматической обработки цифровых изображений и видео. В связи с этим происходит развитие такой области науки, как компьютерное зрение [1].

Технологии компьютерного зрения и системы распознавания образов применяются для решения множества сложных прикладных задач, таких как: распознавание сцены изображения, медицинский анализ, навигация беспилотных автомобилей, анализ и классификация видео, идентификация и аутентификация пользователей. Сегментация – одна из основных задач компьютерного зрения, которая позволяет проанализировать изображения, добиться нужного уровня детализации.

Данная статья посвящена анализу алгоритмов сегментации изображений, основанных на использовании нейронных сетей. Основная идея использования искусственных нейронных сетей состоит в том, чтобы компьютер не только выполнял заранее написанный алгоритм, но и мог обучиться решению поставленной задачи. Также был разработан программный модуль для решения задачи сегментации полупрозрачных объектов с потерей фокусировки на текстовых изображениях с его последующим восстановлением.

1. Полностью свёрточные нейронные сети

Задача сегментации является одной из наиболее сложных, так как в отличие от задачи классификации она требует прогнозирования класса для каждого пикселя входного изображения. Для решения задачи сегментации изображений используются различные модификации

архитектуры свёрточной нейронной сети, которые получили название полностью свёрточные нейронные сети (Fully Convolutional Networks).

Полностью свёрточная нейронная сеть реализует стандартную архитектуру «кодер-декодер», где в качестве кодера используется свёрточная нейронная сеть без полносвязных слоев. Работа кодера заключается в понижении дискретизации изображения, для этого применяются свёрточные слои и слои субдискретизации. Работа декодера заключается в повышении дискретизации изображения, для этого используются транспонированные свертки.

В свою очередь, кодер и декодер связаны skip-соединениями, которые служат для уточнения карт сегментации путем суммирования карт кодера и декодера. В зависимости от количества таких соединений выделяют несколько версий нейронных сетей, а именно [2]:

- FCN-32;
- FCN-16;
- FCN-8.

Схема полностью свёрточной нейронной сети представлена на рис. 1.

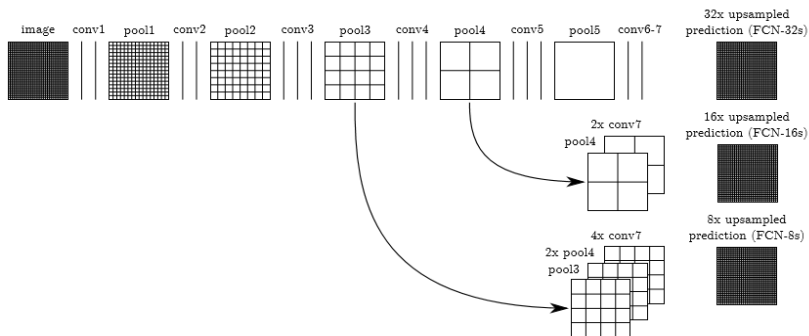


Рис. 1. Схема полностью свёрточной нейронной сети

Преимущества полностью свёрточных нейронных сетей:

- высокая производительность;
- возможность использования предобученной нейронной сети в качестве кодера.

Недостатки полностью свёрточной нейронной сети:

- нечеткие границы сегментированных областей;
- невысокая точность сегментации по современным меркам.

Рассмотрим основные архитектуры искусственных нейронных сетей для сегментации изображений.

2. Архитектура U-Net

U-Net – одна из архитектур полностью свёрточных нейронных сетей, которая реализует схему «кодер- декодер».

Кодер представляет из себя типичную архитектуру свёрточной нейронной сети без полносвязного слоя. Таким образом, кодер состоит из двух повторяющихся слоев свертки с фильтром 3×3 , после которого используется функция активации и операция MaxPooling с шагом 2. На каждом шаге количество каналов увеличивается вдвое.

Декодер состоит из шагов обратной свертки, в ходе выполнения которой количество каналов уменьшается вдвое. Также происходит объединение карт признаков декодера с соответствующей обрезанной картой признаков кодера. Обрезка карт кодера осуществляется в связи с тем, что при каждом выполнении операции свертки, изображение теряет граничные пиксели. На выходе декодера используют свёрточный слой с фильтром 1×1 по всем каналам карт признаков для формирования выходных карт сегментации [3].

Архитектура U-Net представлена на рис. 2.

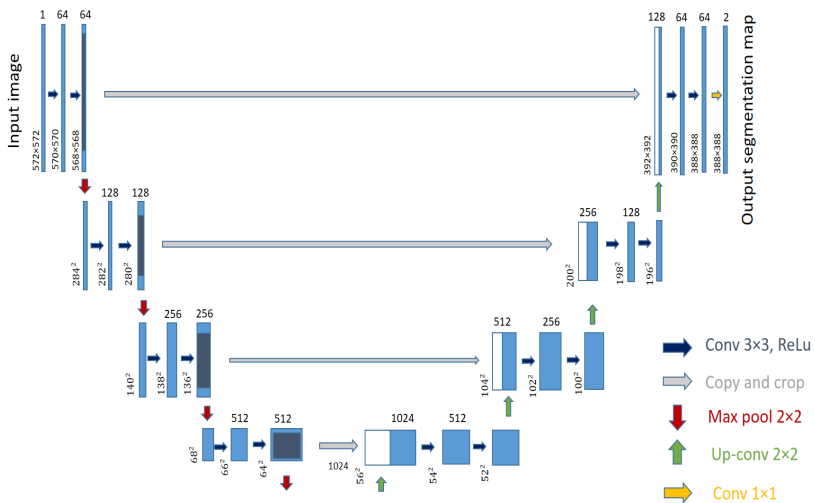


Рис. 2. Архитектура U-Net

Преимущества:

- высокая скорость работы;
- может быть использована не только для сегментации, но и для детектирования объектов на изображениях;

- применяется в большом количестве реальных задач, зачастую для биомедицинских приложений;
- высокое качество карт сегментации даже при небольшой обучающей выборке.

3. Архитектура SegNet

SegNet также является одной из архитектур полностью свёрточных нейронных сетей, основанных на модели «кодер-декодер».

Отличительной чертой данной архитектуры является использование индексов максимального пула с целью повышения частоты дискретизации карт функций с низким разрешением. Это позволяет сохранить высокочастотные детали в сегментированных изображениях и уменьшает количество настраиваемых параметров в декодерах. Архитектура SegNet может быть обучена сквозным использованием стохастического градиентного спуска [4].

SegNet представлена на рис. 3.

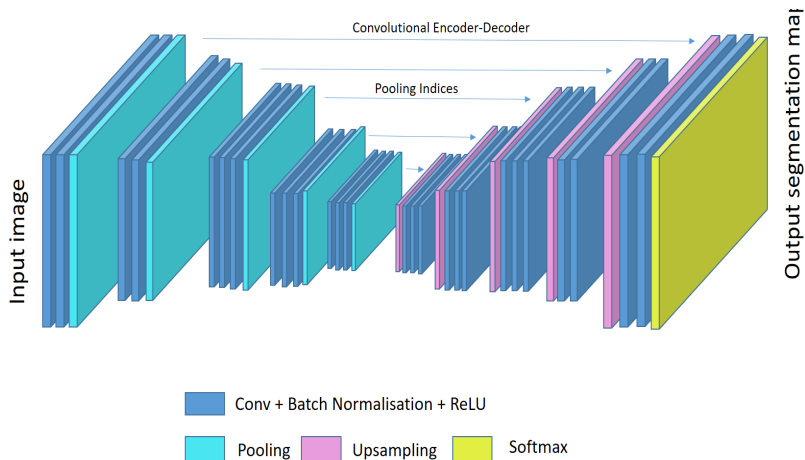


Рис. 3. Архитектура SegNet

Преимущества:

- высокая скорость работы;
- минимальные затраты оперативной памяти компьютера.

Недостатком сети является невысокое разрешение карт сегментации на выходе, то есть невысокая точность сегментации.

4. Архитектура PSPNet

Архитектура PSPNet или «сеть разбора сцены пирамиды» - одна из моделей «кодер-декодер», использующая многомасштабную обработку с применением пирамиды изображений.

Кодер представлен сверточной нейронной сетью ResNet с применением разреженных сверток. Декодер, в свою очередь, использует модуль пулинга пирамиды, в котором происходит многомасштабная обработка изображений с повышающей дискретизацией. На выходе масштабируются карты сегментации до желаемого размера с использованием билинейной интерполяции [5].

Архитектура PSPNet представлена на рис. 4.

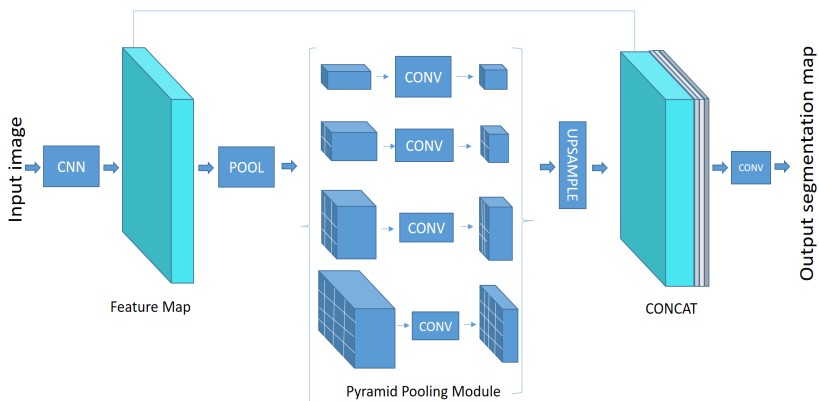


Рис. 4. Архитектура PSPNet

Для получения многомасштабной информации PSPNet применяет 4 различные операции максимального пула с 4 различными размерами окна и шагами. Это эффективно собирает информацию о функциях из 4 различных шкал без необходимости индивидуальной обработки каждого из них

5. Тестирование разработанного программного модуля

Предварительное тестирование разработанного программного модуля производилось на нескольких изображениях одного и того же текста с помехами в виде полупрозрачных объектов с потерей фокусировки, расположенных в различных местах.

Задача заключалась в сегментации помех на изображениях с последующим восстановлением исходного текста. Для решения данной задачи было решено использовать одну из рассмотренных выше

архитектур полностью свёрточных нейронных сетей, а именно, архитектуру U-Net. Была создана обучающая выборка из 15000 изображений, на некоторых из них присутствовали различные полупрозрачные искажения.

Пример распознавания помехи после обучения нейронной сети на тестовом изображении представлен на рис. 5.

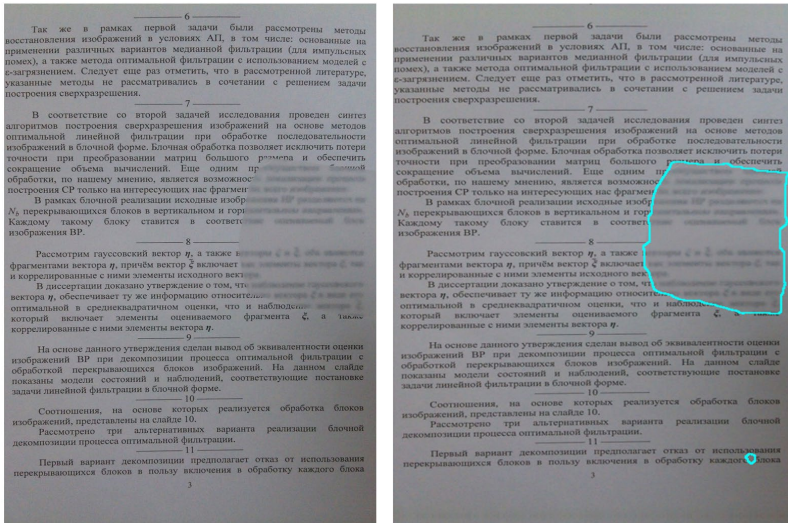


Рис. 5. Сегментация области закрытия

Точность сегментации с использованием архитектуры U-Net составила 96,84%.

После сегментации входных изображений было проведено восстановление исходного кадра. Таким образом, получи очищенное изображение, что показано на рис. 6.

Заключение

Данная статья посвящена разработке и реализации программного модуля для сегментации полупрозрачных помех на текстовых изображениях с его последующим восстановлением. Описаны основные архитектуры нейронных сетей, которые применяются для решения данной задачи. В итоге разработки получился полностью рабочий программный модуль, реализованный в виде библиотеки и тестового приложения. Использование данного решения позволяет улучшать качество текстовых изображений, то есть производить сегментацию

искажений и очищение изображения от различных них. В дальнейшем предполагается использование данного модуля в системе обработки и распознавания текстовых изображений.

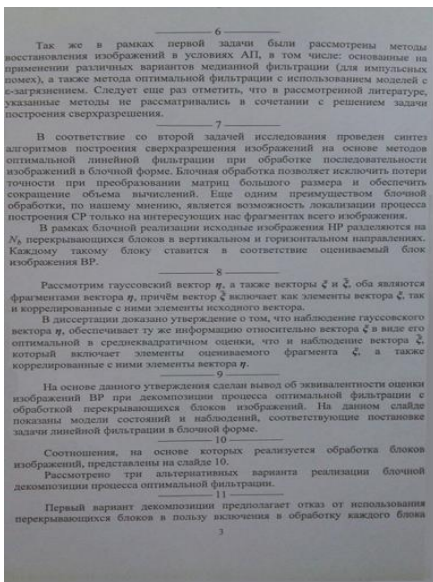


Рис. 6. Очищенное изображение

Список литературы

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
2. Fully convolutional networks for semantic segmentation [Электронный ресурс]. – Режим доступа : <https://arxiv.org/pdf/1605.06211v1.pdf/>
3. U-Net: Convolutional Networks for Biomedical Image Segmentation [Электронный ресурс]. – Режим доступа : <https://arxiv.org/pdf/1505.04597.pdf/>
4. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation [Электронный ресурс]. – Режим доступа : <https://arxiv.org/pdf/1511.00561.pdf/>
5. Pyramid Scene Parsing Network [Электронный ресурс]. – Режим доступа : <https://arxiv.org/pdf/1612.01105v2.pdf/>

Разработка радиоприёмника на базе микросхемы модели RDA5807

В. А. Соколов

Студент бакалавриата

С. А. Зуев

Доцент

Введение

В эру современных технологий существует большое количество способов коммуникации между людьми. Каждый существует для решения определённых задач. Некоторые уже утратили свою актуальность, больше нет нужды в их постоянном использовании. Но всё ещё существуют устройства, которые невозможно заменить в виду применяемых качества работы и стабильности в них технологий по сравнению с более новыми альтернативами. Речь идёт о радиоприёмниках. Со временем их роль меняется, меняется схемотехника, но их устройство остаётся прежним [1, 2].

Под радиоприёмным устройством понимают радиоприёмник, снабженный антенной, а также средствами обработки принимаемой информации и воспроизведения её в требуемой форме (визуальной, звуковой, в виде печатного текста и т. п.). Во многих случаях антенна и средства воспроизведения конструктивно входят в состав радиоприёмника [3]. Радиоприёмное устройство выполняет пространственную, временную и поляризационную селекцию радиоволн и их преобразование в электрические радиосигналы (напряжение, ток) с помощью антенны. Также происходит преобразование по частоте, выделение полезного радиосигнала из совокупности других (мешающих) сигналов и помех, действующих на выходе приёмной антенны и не совпадающих по частоте с полезным сигналом, усиление, преобразование полезного радиосигнала к виду, позволяющему использовать содержащуюся в нём информацию. Формально радиоприёмные устройства относят к радиостанциям, хотя такая классификация редко встречается на практике.

1. Реализация

Для начала необходимо определиться с основным модулем, на базе которого будет выполнен радиоприёмник. В данном случае, был выбран

модуль RDA5807, представленный на рис. 1. Характеристики модуля представлены в таблице.

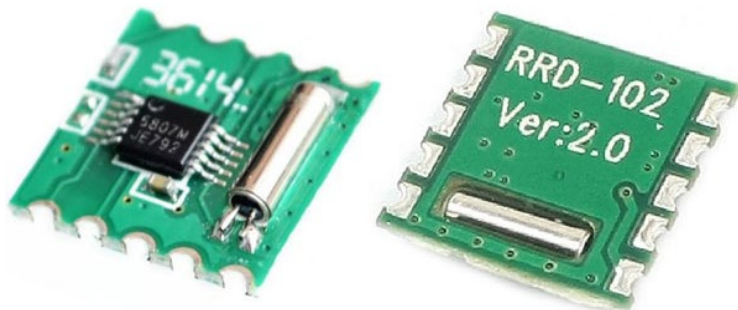


Рис. 1. Внешний вид модуля RDA5807

Таблица

Характеристики модуля

Общие параметры	Значение
Диапазон частот	76 – 108МГц
Шина передачи данных	I2C (максимальная частота – 400 кГц)
Возможность переключения звука	Сtereo/Моно
Отключение плавное звука и регулировка громкости	есть
Управление басами	есть
Режим ожидания	есть
Встроенный LDO регулятор	есть
Настраиваемый шаг между каналами	200 кГц, 100 кГц, 50 кГц, 25 кГц
Индикатор уровня сигнала	есть
Корпус MSOP	10 выводов
Напряжение питания	3 вольта (от 1,8 до 3,3 вольт)
Ток потребления в рабочем режиме	до 21 мА
Ток потребления в спящем режиме	5 мкА
Коэффициент нелинейных искажений	0,15 – 0,2 %

Для того чтобы реализовать радиоприёмник также необходимы следующие компоненты: микроконтроллер Arduino Pro Mini, усилитель

с динамиками, дисплей, аккумулятор 18650, контроллер заряда и корпус из оргстекла [4]. Элементы устройства приведены на рис. 2-5.

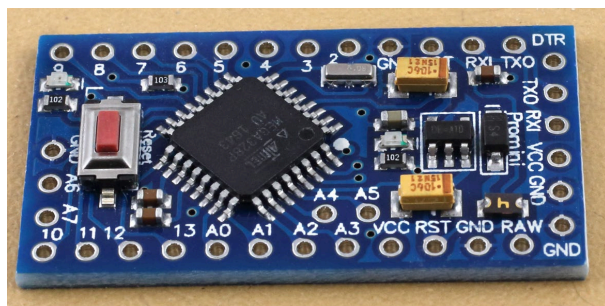


Рис. 2. Микроконтроллер Arduino

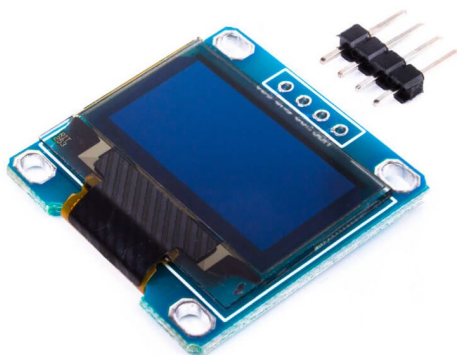


Рис. 3. Дисплей

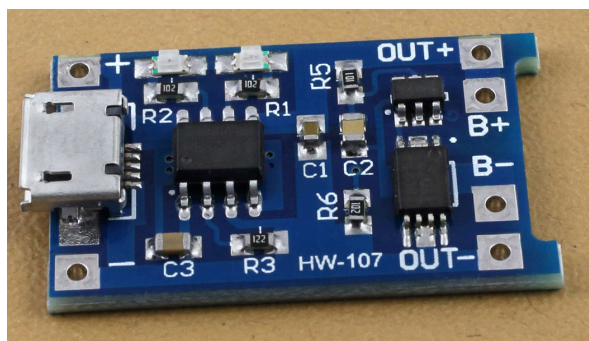


Рис. 4. Контроллер заряда

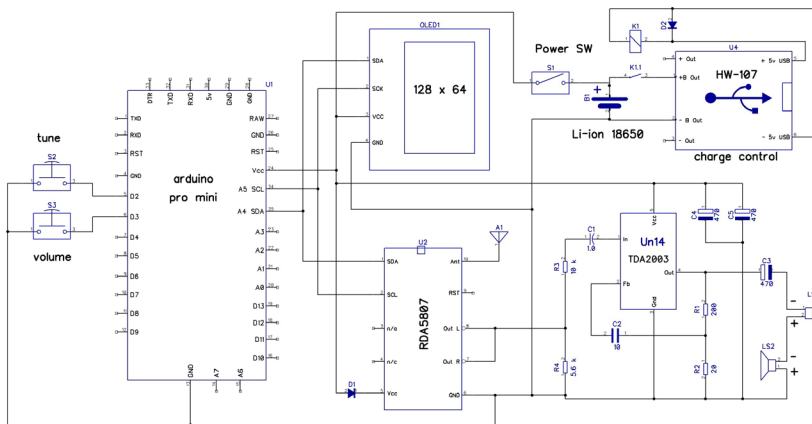


Рис. 5. Полная электрическая схема

Важно заметить, что данный модуль не перегорает от 5 вольт, но на более высоком напряжении модуль прекращает работу или возникает шум, вместо ожидаемых радиостанций. Данный модуль необходимо питать через стабилизатор K142EN на 3.3 вольта. При напряжении ниже данного значения, радиоприёмник перестаёт реагировать на нажатие кнопок, так как свою работу прекращает Arduino. Но независимо от этого сам модуль продолжает держать заданную частоту при неизменном качестве.

2. Написание скетча

Для написания применяются стандартные библиотеки SSD1306ASCII для дисплея и RDA5807M для радио [5]. Строка RDS содержит в себе некоторую информацию о радиостанции, а именно присутствуют три блока: текущее время, имя радиостанции и собственно сама информационная строка. Кроме названия радиостанции, дисплей показывает текущую силу сигнала, громкость и рабочую частоту.

Возможна реализация памяти для запоминания рабочих частот. В текущей реализации присутствуют три кнопки: первая служит для переключения режимов меню. Две других кнопки используются для увеличения или уменьшения выбранного в меню параметра. Код хранится в [6].

Два динамика, направленные в противоположные стороны, работают на громкости достаточной для комфортного прослушивания радиостанций.

Применяется готовый корпус из оргстекла, внутри которого размещаются компоненты радиоприёмника. Существует возможность производства корпуса с применением 3D печати.

Заключение

Реализация радиоприёмника на основе описанных выше компонентов не является столь затратной, как в плане ресурсов, так и денежном эквиваленте. Все компоненты легко найти, а общая стоимость проекта не превышает разумных пределов для этого столь высокотехнологичного устройства. Создание такого радиоприёмника полезно для практики в области электроники, и в результате было получено очень полезное в жизни устройство.

Список литературы

1. Палшков, В. В. Радиоприёмные устройства / В. В. Палшков. – М. : Издательство «Радио и связь», 1984. – 320 с.
2. Радиоприёмник [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%BE%D0%BF%D1%80%D0%B8%D1%91%D0%BC%D0%BD%D0%B8%D0%BA>
3. Буга, Н. Н. Радиоприёмные устройства / Н. Н. Буга, А. И. Фалько, Н. И. Чистяков; под ред. Н. И. Чистякова. – М. : Издательство «Радио и связь», 1986. – 320 с.
4. Еще одно FM радио на RDA5807 под управлением Ардуино [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/post/485448/>
5. Цифровой FM приёмник на Arduino и модуле RDA5807 [Электронный ресурс]. – Режим доступа : <https://full-chip.net/arduino-proekty/97-cifrovoy-fm-priemnik-na-arduino-i-module-rda5807-s-graficheskim-displeem-i-funkciey-rds.html>
6. Репозиторий GitHub [Электронный ресурс]. – Режим доступа : <https://github.com/CuteSergal/ArduinoRadioDemo>

Разработка модуля «WWW» для web-интерфейса SOHO-сервера

С. В. Стратиенко

Студент бакалавриата

А. А. Головкин

Ассистент

Введение

В настоящее время под домашней или небольшой офисной сетью понимается локальная сеть, объединяющая компьютеры, принтеры, сканеры, цифровые камеры, и другое оборудование. Устройства пользователей чаще всего располагаются в одной квартире или офисе. Данный сегмент рынка электроники, предназначенный для домашнего использования, называется SOHO (от англ. Small Office, Home Office) [1].

Ресурсы класса SOHO как правило характеризуют устройства, не предназначенные для производственных нагрузок и довольно хорошо переживающие длительные периоды бездействия. Долгие годы считалось, что эта сугубо офисная технология малопригодна для домашних сетей и подключения к Интернету. Однако благодаря дешевизне и простоте решений она стала повсеместно использоваться.

Сеть SOHO – локальная компьютерная сеть. Сети SOHO являются наиболее простым и доступным способом реализации локальной сети в квартире. В сети используется сетевые коммутаторы Ethernet, повторители, или беспроводная сеть Wi-Fi. Беспроводные решения стали мощным фундаментом для создания «умных домов», обеспечивая более гибкие решения по сравнению с альтернативными технологиями. Сеть позволяет использовать ресурсы всех компьютеров для передачи/хранения данных, а также получать доступ в сеть Интернет через один из компьютеров или сетевой шлюз. В сети SOHO можно использовать сервер для контроля доступа к сети, общего хранилища данных, а также разделять права пользователей. SOHO–серверы входят в состав одноименного типа сети и предназначены для разностороннего обслуживания небольшого числа рабочих станций [2].

Представленная в статье работа посвящена разработке и программной реализации модуля “WWW” для web–интерфейса, который будет способен обеспечить интуитивно понятную настройку

компонентов, необходимых для деятельности в сети рабочих станций и устройств. Схема построения SOHO-сети представлена на рис. 1.

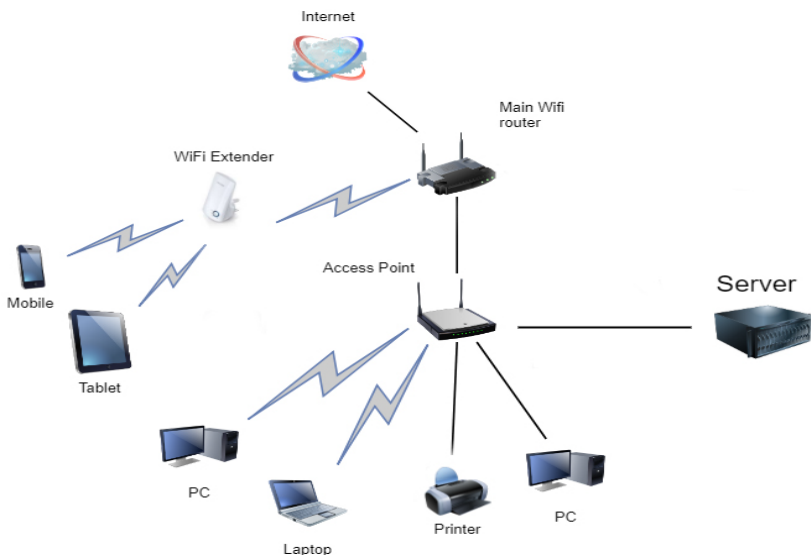


Рис. 1. Пример схемы построения сети класса SOHO

2. Реализация модуля “WWW”

Рассмотрение следует начать с разъяснений относительно используемых при разработке технологий.

Круг задач, решаемых Web -сервером, ограничен. В основном он сводится к поддержке HTTP-взаимодействия и доставке клиенту Web -документов. Любые «нестандартные» действия реализуются с помощью специальной программы, которая взаимодействует с веб-сервером и клиентом. Это взаимодействие подчиняется определенным правилам.

Основной набор таких правил – стандарт CGI (Common Gateway Interface – интерфейс общего шлюза), который определяет порядок запуска программы на компьютере-сервере, способы передачи программе параметров и доставки результатов ее выполнения клиенту [3].

Программу, которая работает по интерфейсу CGI совместно с Web-сервером, принято называть скриптом. Как правило, скрипт представляет собой исполнимый файл, который принимает запрос от Web-сервера, выполняет необходимые операции, формирует результат в виде HTML страницы и передает результаты запроса серверу.

На рис. 2 изобразим диаграмму последовательности взаимодействия пользователя с веб-сервером. Так как данные диаграммы вполне наглядны и понятны, из них сразу же можно понять какие выполняются действия и какова их последовательность: веб-приложение анализирует файл конфигурации, и на основе анализа создает пользовательскую HTML-форму. С помощью этой формы администратор может изменять параметры сервера. После отправки формы, веб-приложение проводит его валидацию, и, если данные допустимы, проводит непосредственную настройку сервера путем правки файла конфигурации.

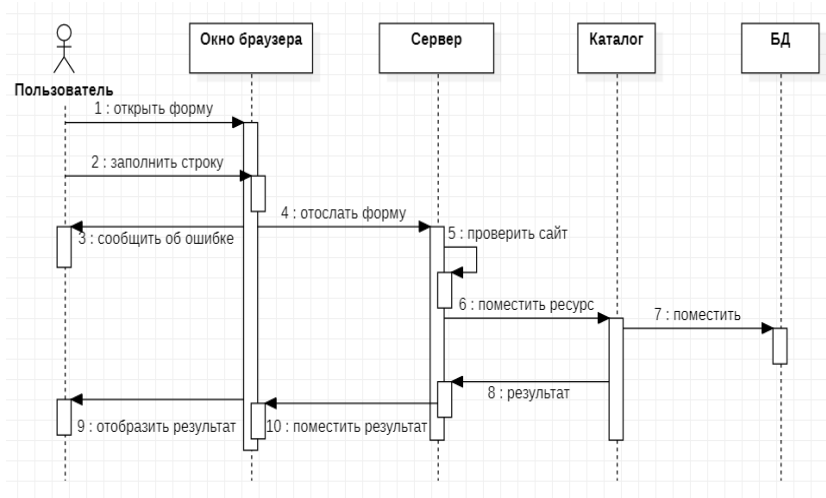


Рис. 2. Диаграмма последовательностей взаимодействия

В качестве средств реализации для разработки модуля, были выбраны следующие средства:

- Python 3.6.9 – язык программирования;
- Flask 1.1.1 – фреймворк для создания вебсайтов на языке Python;
- Nginx – web-сервер для реализации выполнения скриптов;
- Ubuntu 18.04.4 LTS – дистрибутив Linux;
- Oracle VM VirtualBox – виртуальная машина для установки Linux в качестве дополнительной системы [4, 5].

На рис. 3 представлен интерфейс, который позволяет создать домен по указанному URL и выбрать путь до корневой веб-страницы HTML. После заполнения и отправки пользовательской формы

администратором, домен будет доступен на конфигурируемом сервере по заданному URL.

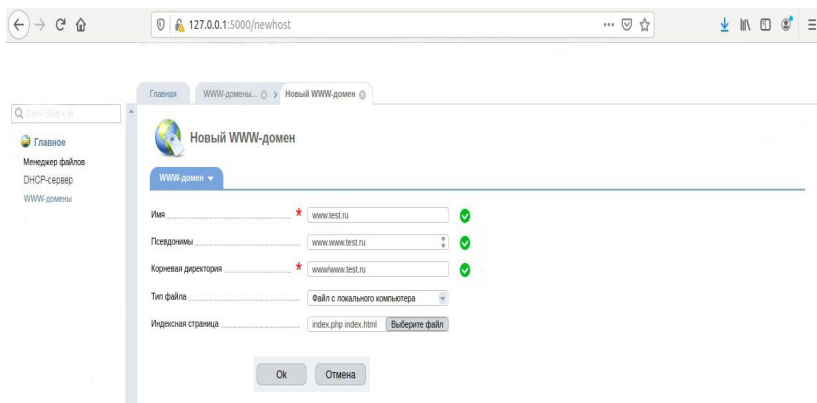


Рис. 3. Создание нового домена “WWW” на сервере

На рис. 4 представлен интерфейс, который показывает список доменов и список поддоменов нашего сервера. Здесь можно создать новый домен, изменить или удалить уже существующие, а также перейти на интересующий нас домен. На рис. 5 представлен переход в менеджер файлов, где можно добавить новые файлы для работы домена.

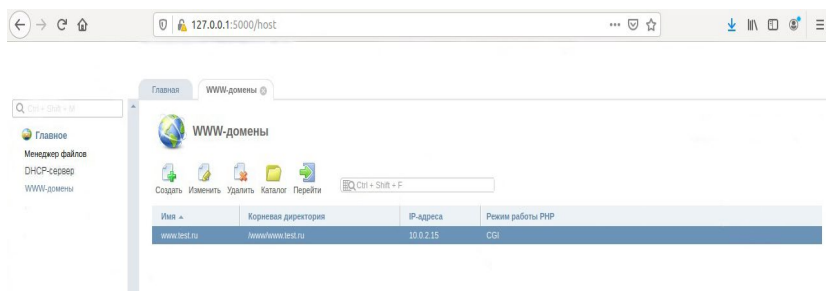


Рис. 4. Список доменов “WWW” на сервере

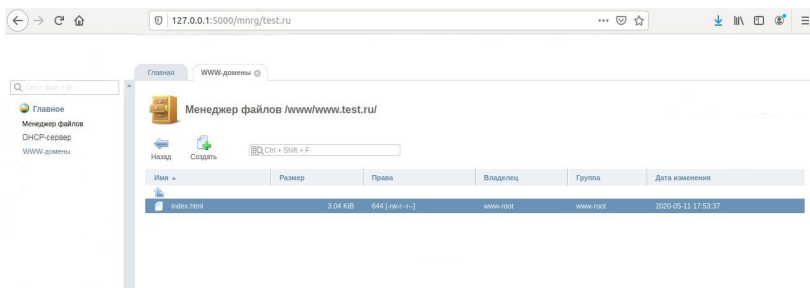


Рис. 5. Менеджер файлов локального веб-сервера

Заключение

В данной статье рассмотрен вариант реализации модуля «WWW» для web-приложения, который позволяет конфигурировать сервер класса SOHO. Рассмотрен отдельный процесс взаимодействия пользователя с веб-сервером. Предложенный алгоритм был реализован на базе компьютера с ОС Linux. Использование данного решения позволяет создать локальный домен для специфических задач. В дальнейшем список охватываемых параметров предполагается расширить в целях повышения функциональности и востребованности данного решения.

Список литературы

1. Журнал ИКС [Электронный ресурс]: электронный науч. журн. – Режим доступа : <http://www.iksmedia.ru/articles/2211-Seti-SOHO.html>
2. Телеком системы [Электронный ресурс] : электронный науч. журн. – Режим доступа : <https://tele78.ru/tekh-materialy/soho-seti>
3. Олифер В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – 6-е изд. – Санкт-Петербург : Питер, 2016. – 759 с.
4. Уорд Б. Внутреннее устройство Linux / Б. Уорд. – Санкт-Петербург : Питер, 2016. – 384 с.
5. Unix и Linux. Руководство системного администратора / Э. Немет [и др.]. – 5-е изд. – М. : «ДИАЛЕКТИКА», 2020. – 1168 с.

Система предобработки выбора контекстно-свободных грамматик для задач Text Mining

Е. П. Ступникова

Студент бакалавриата

В. В. Гаршина

Доцент

Введение

Экспоненциальный рост объема информации является причиной все более и более возрастающей трудности поиска необходимых документов и организации их в виде структурированных по смыслу баз данных. Известно, что около 80 процентов важной для бизнеса информации существует в неструктурированной текстовой форме. Технология Text mining позволяет извлечь из этих данных ценные знания, такие как факты или бизнес-правила, которые нельзя получить в результате обычной обработки объема данных.

Технология имеет широкий спектр сфер использования. Можно выделить корпоративную бизнес-аналитику в таких задачах, как например сканирование набора документов, составленных на естественных языках для упрощения поиска по базе данных или оценка интересов пользователей и таргетированная реклама. Так же технология широко используется в национальной безопасности, научных исследованиях, анализе рынков и социальной обстановки.

Тем не менее, если задачи поиска структурированной информации в большинстве случаев оказываются решены, то задачи анализа текста, написанного на естественном языке, имеют массу проблем. Дело в том, что для улучшения качества работы методы TextMining должны быть адаптированы к предметной области. При этом многие типы текстов плохо поддаются машинной обработке.

1. Цели и задачи работы

Цель работы – реализация системы предобработки выбора контекстно-свободных грамматик, в основе которой лежит классификатор, обеспечивающий на основе результатов разбора предложения синтаксическим парсером выбор необходимой для выделения фактов формальной грамматики. Необходимо изучить предметную область, а так же инструменты для реализации. После чего

- проанализировать имеющиеся данные, разработать классификатор, обучить его на имеющихся данных и провести тестирование.

2. Анализ предметной области

Технология Text Mining представляет собой одну из разновидностей методов Data Mining и подразумевает процессы извлечения знаний высококачественной информации из текстовых массивов. Это происходит посредством выделения шаблонов обычно с помощью средств статистического изучения [1].

Статистические методы позволяют достигнуть определённых результатов в решении ряда задач, например, распознавание речи. Однако перспективным является использование гибридных моделей, например, использование машинного обучения в этой сфере. Примерами могут быть классификация текстовых документов или задача построения предметной онтологии на основе текста.

В Text Mining используется математическая модель реального языка – формальный язык. Содержание конкретной математической модели зависит от того, какие свойства важны для изучения, т.е. что планируется в данный момент выделить и изучать.

Такой формальный язык задаётся с помощью грамматик, описывающих законы внутреннего строения его цепочек. Чаще всего используются контекстно-свободные и регулярные грамматики. Грамматика называется контекстно-свободной, если в левой части правил её вывода стоит символ из множества нетерминальных (VN), а справа – любой символ из множества нетерминальных и терминальных (VN):

$$A \rightarrow \beta, \text{ где } A \in VN, \beta \in (VT \cup VN) \quad (1)$$

Регулярные грамматики описываются в виде регулярных выражений и применяются как шаблоны текстового поиска.

Сам процесс распознавания высказывания называется парсинг. Строка разбивается на набор символов, и каждый из них анализируется грамматикой языка.

Таким образом существует 2 способа реализации извлечения фактов, в частности, именованных сущностей из текста.

Первый способ - методы, основанные на грамматиках лингвистических моделей. В таких методах с помощью правил контекстно-свободных грамматик парсеры (например Яндекс Томита) могут извлекать структурированную информацию, в частности именованные сущности. Недостатки такого подхода очевидны: чтобы покрыть правилами достаточное количество языковых ситуаций, требуются сложные правила и люди, которые смогут их писать и

поддерживать. Точность может быть высокой, а вот полнота (покрытие) — нет. К преимуществам можно отнести достаточно высокую скорость обработки текстов.

Второй способ реализации извлечения фактов - машинное обучение. Для решения задач Text Mining используются модели классического машинного обучения, работающие с последовательностями. Для обучения таких моделей требуется для каждой цепочки токенов из обучающей выборки составить цепочку векторов-признаков [2].

3. Структура проекта и средства реализации

Разрабатываемый проект является частью более большого проекта по извлечению фактов из неструктурированных текстов с целью использования их в конечных приложениях пользователя. В результате увеличения объёма текстов и используемых грамматик, возникла необходимость в создании классификатора, получающего на вход предложение и отвечающий, какую грамматику необходимо использовать для разбора. Структура проекта показана на рис. 1.



Рис. 1. Структура проекта

Для реализации классификатора выбран язык Python. Используются такие программные пакеты как pandas, numpy, scikit-learn и Tensorflow. Проблема выделения фактов из текста решается применением Томита-парсера – свободно распространяемого инструмента от Яндекс, который получает на вход текст на естественном языке и далее, с помощью словарей и грамматик, составленных пользователем, он преобразует этот текст в набор структурированных данных [3].

4. Подготовка данных

Для достижения требуемого вида формализации понадобилось конвертировать файлы в формате json, в котором представляется выходная разметка шаблонов грамматик в Томита-парсере, в xml, а из файла xml с помощью написанного на языке Microsoft Visual Basic макроса сформированы индивидуальные txt файлы, в которые были изъят исключительно тексты предложений. Файлы с предложениями были рассортированы по четырём папкам с идентификаторами грамматик. Таким образом объём выборки составил 1000 предложений, из которых 80% использовалось для обучения сети и 20% для тестирования.

5. Программная реализация

Разработанная нейронная сеть представлена на рис. 2. Она состоит из входного слоя, трёх полносвязных слоёв с функцией активации relu и последним слоем с функцией активации soft max. Для первых двух слоёв используется drop out (отключение нейронов) для избегания проблемы переобучения.

Нейронная сеть была обучена в течение 20 эпох с размером пакета (batch size), равным 32 [4].

В результате тестирования было выявлено, что модель имеет ассигасу в 0.85

Заключение

Результатом проведенного исследования стала разработка и программная реализация системы предобработки выбора контекстно-свободных грамматик, в основе которой лежит классификатор, обеспечивающий на основе результатов разбора предложения синтаксическим парсером выбор необходимой для выделения фактов формальной грамматики. Была изучена предметная область и описаны способы извлечения именованных сущностей из текста. В итоге была разработана нейронная сеть, производящая предобработку массива контекстно-свободных грамматик и позволяющая определить, какую грамматику необходимо использовать для конкретного предложения в Томита-парсере. Данная разработка значительно упрощает работу по извлечению необходимых фактов из текста, ускоряя использования парсера.

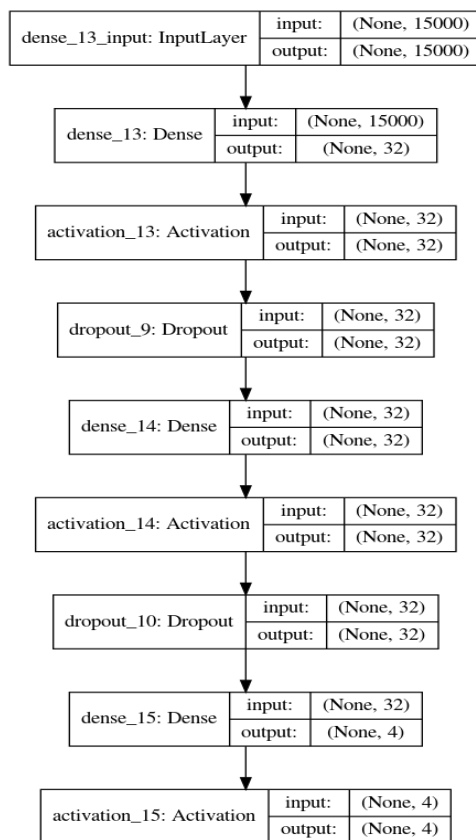


Рис. 2. Архитектура нейронной сети

Список литературы

1. Богатырёв, С. Введение в добычу данных (Data Mining) / С. Богатырёв, А. Симонова. – М. : БИНОМ, 2006. – 34 с.
2. Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP / Барсегян А.А. [и др.]. – СПб. : БХВ-Петербург, 2004. – 306 с.
3. Томита-парсер. Руководство разработчика. [Электронный ресурс]. – Режим доступа : <https://tech.yandex.ru/tomita/doc/dg/concept/about-dospage/>
4. Keras : The Python Deep Learning library [Электронный ресурс] – Режим доступа : <http://www.keras.io>

Управление динамикой системы Малкуса–Лоренца посредством гистерезисных преобразователей

М. А. Теплякова

Студент бакалавриата

М. Е. Семёнов

Профессор

Введение

Работа Эдварда Лоренца [1], опубликованная в 1963 году, заложила основы теории хаоса. В данной работе Лоренца из системы уравнений Навье–Стокса впервые была получена приведённая ниже нелинейная автономная система (1) обыкновенных дифференциальных уравнений третьего порядка, описывающая конвекцию в плоском слое жидкости постоянной толщины.

Позднее было доказано, что данная система уравнений применима не только к задаче о конвекции в слое жидкости, но и к другим физическим системам. К ним относятся: одномодовая модель лазера, конвекция в трубе, диссипативный осциллятор с инерционным возбуждением.

К моделям, описываемым системой уравнений Лоренца, также относится модель водяного колеса, что было доказано Малкусом [2]. Данная модель представляет собой колесо, на ободе которого закреплены чаши с отверстиями на дне. Сверху на колесо, симметрично относительно оси вращения, льётся поток воды, и, в зависимости от скорости этого потока, колесо будет оставаться неподвижным, вращаться с постоянной скоростью, либо вращаться хаотически.

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases} \quad (1)$$

В приведённой выше системе безразмерный параметр x отвечает угловой скорости колеса, y и z – координаты центра масс рассматриваемой системы.

Классические значения параметров, при которых как правило производятся численные эксперименты, для данной системы следующие:

$$\sigma = 10, r = 28, b = 8/3 \quad (2)$$

Решением системы при данных параметрах является странный аттрактор (рис. 1).

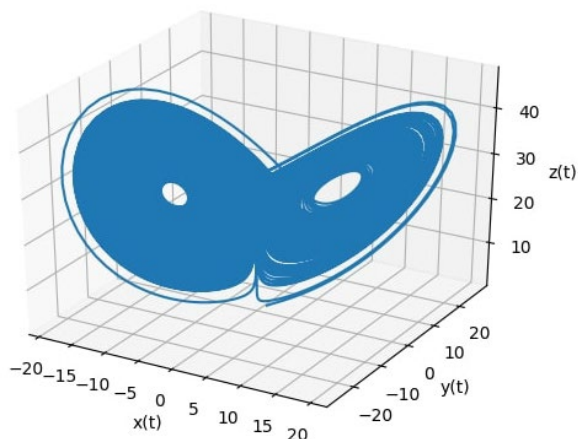


Рис. 1. Фазовый портрет системы Лоренца при классических значениях параметров

Однако, стоит отметить, что классическая система Лоренца не учитывает трение на ободе колеса, имеющее место в реальных физических моделях. В данной работе предложено моделирование данного физического процесса с помощью гистерезисного преобразователя типа люфт, проведено исследование модифицированной системы и сравнение полученной модифицированной системы с классической системой Малкуса–Лоренца.

2. Модифицированная система Малкуса–Лоренца с гистерезисным преобразователем типа люфт

Моделирование трения далее реализовано с помощью гистерезисного преобразователя типа люфт [3]. Модифицированная система Малкуса–Лоренца будет иметь следующий вид:

$$\begin{cases} \dot{x} = \sigma(y - x) - M \cdot \text{sign}(\dot{x}) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases} \quad (3)$$

Физический смысл переменных x , y и z остаётся неизменным. Параметр M в данной системе отвечает коэффициенту трения, L – оператор люфта. Действие оператора люфта можно представить следующим образом. Обозначим аргумент функции $sign$ как функцию $n(t)$.

$$n(t) = L_0[t_0, x(t_0)] \cdot x(t) \quad (t \geq t_0) \quad (4)$$

В данном случае $x(t)$ – подаваемое на вход люфта значение, $x(t_0)$ – фиксированное начальное состояние системы, $n(t)$ – выход люфта. Выход люфта будет определяться следующим равенством:

$$n(t) = \begin{cases} x(t_0), \forall t: x(t_0) - \frac{h}{2} \leq x(t) \leq x(t_0) + \frac{h}{2} \\ x(t), \forall t: x(t) \geq x(t_0) + \frac{h}{2} \\ x(t) + h, \forall t: x(t) \leq x(t_0) - \frac{h}{2} \end{cases} \quad (5)$$

где h – раствор люфта. Графически зависимость выхода люфта от его входа можно изобразить следующим образом (рис. 2). В ходе численного эксперимента был также получен фазовый портрет для модифицированной системы Малкуса–Лоренца с данным гистерезисным преобразователем (рис. 3).

Для того чтобы провести качественный анализ поведения модифицированной системы Малкуса–Лоренца в сравнении с поведением классической системы Малкуса–Лоренца, были построены следующие бифуркационные диаграммы (рис. 4, 5) по параметру z с изменяющимся параметром r . Данные диаграммы могут дать представление о смене динамических режимов в системе; из представленных ниже диаграмм можно сделать вывод о том, что при выбранных параметрах модифицированная система с гистерезисным преобразователем переходит в состояние хаоса при большем значении параметра r в сравнении с классической системой Малкуса–Лоренца. Сценарий перехода к хаосу при увеличении параметра r также изменяется (рис. 5) и требует проведения детального анализа.

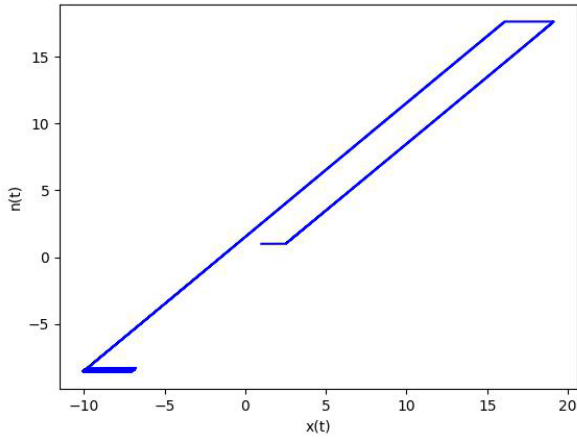


Рис. 2. Выход люффа, раствор люффа $h = 3$

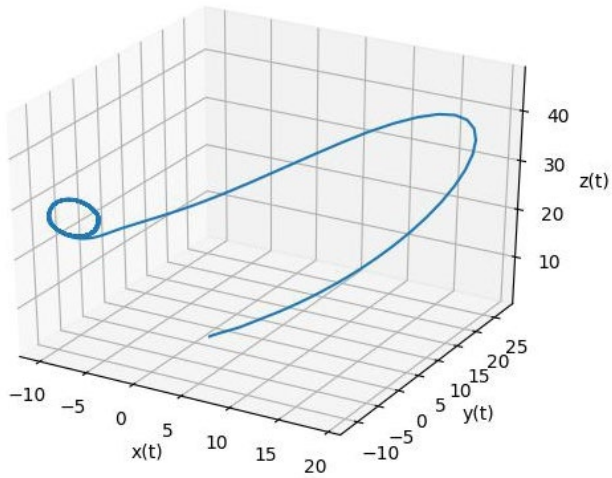


Рис. 3. Фазовая траектория модифицированной системы Лоренца при классических параметрах и $M = 15$, $h = 3$

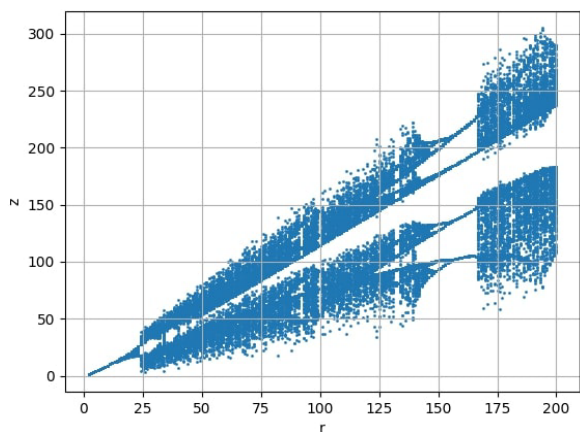


Рис. 4. Бифуркационная диаграмма для системы Малкуса–Лоренца при классических параметрах

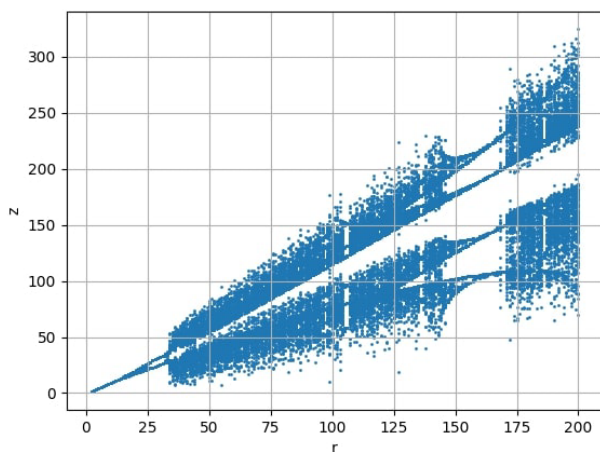


Рис. 5. Бифуркационная диаграмма для модифицированной системы Малкуса–Лоренца при классических параметрах и $M = 15$, $h = 3$

Заключение

В данной работе было проведено исследование модифицированной системы Малкуса–Лоренца с гистерезисным преобразователем типа люфт, полученной при рассмотрении модели водяного колеса с трением на ободе.

Были построены бифуркационные диаграммы, позволяющие провести качественное сравнение поведения классической и модифицированной системы Лоренца. Сценарий смены динамических режимов полученной модифицированной системы требует дальнейшего исследования.

Список литературы

1. Lorenz, E. N. Deterministic nonperiodic flow / E. N. Lorenz // *Journal of the Atmospheric Sciences*. – 1963. – Vol. 20(2). – P. 130-141.
2. Красносельский, М. А. Системы с гистерезисом / М. А. Красносельский, А. В. Покровский. – М. : Наука, 1983. – 272 с.
3. Malkus, W. V. R. Non-periodic convection at high and low Prandtl number / W. V. R. Malkus // *Mémoires de la Société Royale des Sciences de Liège*. – 1972. – Tome IV. – P. 125-128.

Применение нейронных сетей для генерации мелодий

М. В. Толстов

Студент бакалавриата

Д. И. Соломатин

Старший преподаватель

Введение

Нейронные сети используются для улучшения всех областей нашей жизни. Одной из популярных в последнее время тем, связанных с искусственным интеллектом, является обучение без учителя (unsupervised learning). Одной из задач, решаемых обучением без учителя, является генерация контента. Контентом могут быть изображения, текст или музыка.

В данной статье рассматривается задача генерации музыкальных записей при помощи методов глубинного обучения.

1. Возможные варианты архитектуры нейронной сети

Первая волна работ, связанных с генерацией музыки началась в 1988 году. Дж. П. Льюис (J P Lewis) и Питер М. Тодд (P M. Todd) предложили использовать нейронные сети для автоматического создания музыки. Льюис в своей работе [1] использовал многослойный перцептрон, а Тодд в [2] применил авторегрессионную нейронную сеть

(Recurrent neural network, RNN) – способ, который после стольких лет остался актуальным.

Реализованная модель основана на LSTM – слоях, которые являются модификацией архитектуры RNN.

Главное отличие традиционных нейронных сетей от человека заключается в том, что они начинают с чистого листа каждый раз, когда им необходимо решить какое-либо задание. Так, например, во время чтения человек понимает значение последнего прочитанного слова, опираясь на предшествующие ему. И если традиционные нейронные сети плохо подходят для решения задач, зависящих от знания каких-либо предшествующих данных, то рекуррентная нейронная сеть (RNN) как раз предназначена для их выполнения.

В рекуррентной нейронной сети есть циклы, передающие данные от шага к шагу. Более того, её можно «развернуть», то есть представить, как показано на рис. 1, в виде цепи, состоящей из одной и той же нейронной сети.

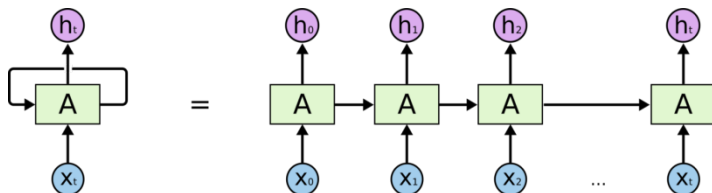


Рис. 1. Архитектура RNN

В тех случаях, когда для работы сети не требуются длительные последовательности, рекуррентная сеть показывает себя отлично. Однако проблема заключается в том, что при решении некоторых задач для предсказания нужен анализ, более длительного контекста.

Теоретически можно подобрать такие параметры, при которых рекуррентная нейронная сеть могла бы работать с очень длительной последовательностью. Но фактически она не может обучиться долговременным зависимостям, так как страдает краткосрочной памятью. Это происходит из-за проблемы затухающего градиента: слои, до которых доходит градиент, близкий к нулю, практически перестают обучаться, а в случае с рекуррентными нейронными сетями это слои, обрабатывающие предшествующие данные.

В 1997 году была предложена концепция долгой краткосрочной памяти (Long Short Term Memory, LSTM) – разновидность рекуррентной нейронной сети. Она была специально разработана для работы с долговременными зависимостями. LSTM (рис. 2) имеет схожую

структуру с рекуррентной нейронной сетью, но в ней присутствует внутренние механизмы, названные шлюзами (gates), которые регулируют потоки информации.

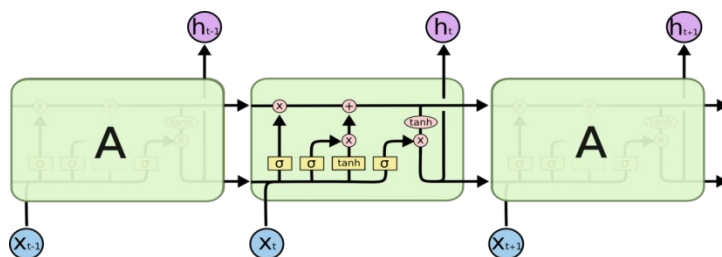


Рис. 2. Архитектура LSTM

Эти шлюзы учатся распознавать, какая информация из последовательности важна, а какая – нет.

Ключевая идея LSTM – состояние ячейки, которое можно назвать “памятью” сети. Состояние ячейки – переносчик информации на всем протяжении последовательности, с которой работает нейронная сеть, поэтому теоретически информация о первых элементах последовательности может сохраниться до конца. По мере перемещения состояния ячейки в нее добавляется и удаляется информация. Шлюзы (gate) – это разные нейронные сети, во время тренировки они способны выучить то, какую информацию необходимо запомнить или забыть.

В архитектуре, показанной на рис. 2, присутствует 3 шлюза:

- шлюз забывания;
- входной шлюз;
- выходной шлюз.

Шлюз забывания отвечает за то, какую информацию необходимо оставить в состоянии ячейки, на выходе из этого шлюза все значения находятся в интервале от 0 до 1, что приводит к тому, что при поточечном умножении с состоянием ячейки значения, близкие к 0, приводят к забыванию информации.

Входной шлюз, в свою очередь, отвечает за добавление новой информации в состояние ячейки, опираясь на текущий элемент последовательности.

Выходной шлюз выполняет функцию некоего фильтра состояния ячейки, так как он принимает решение, какую информацию необходимо передать на выход, при этом опираясь на текущий элемент последовательности. См. об этом в [3].

2. Реализация

Рассматривая возможность использования рекуррентной нейронной сети для генерации музыкальных записей при помощи методов глубинного обучения, необходимо отметить следующее:

Для обучения нейронной сети использовано символьное музыкальное представление (SMR). Согласно [4] нота или аккорд могут быть заданы при помощи частоты, длительности, громкости. Одним из протоколов, позволяющих хранить записи в данном формате, является Musical Instrument Digital Interface (MIDI [5]). Согласно ему записи представляются в виде последовательностей команд, каждой из которых соответствует время её исполнения. Основные команды: «Note on» и «Note off», указывают на начало и конец воспроизведения определенных нот соответственно.

В рамках данной работы, для обучения была взята фортепианная музыка – около 90 саундтреков игры «Final Fantasy» в формате midi.

В результате обработки данных было получено 57177 нот и аккордов, из них 352 – уникальные. Наиболее распространенные длительности нот – восьмая такта (23825, ~42%) и шестнадцатая такта (10048, ~18%). Опираясь на то, что в выборке большинство нот были восьмые, было предположено, что при обучении нейронной сети и создании новых треков длительность нот не является ключевой и может быть опущена. При этом при генерации музыки было принято решение считать по умолчанию, что все ноты имеют размер, равный восьмой такта.

Подготовка данных для обучения состояла из трех этапов, подобных тем, что были использованы в [6]: преобразования в более простой формат, токенизации и разбиения на примеры (сэмплирование). На первом этапе последовательности нот и аккордов были преобразованы в списки строк, в которых каждой ноте соответствует её номер в спецификации midi, а аккорду – последовательность чисел, разделенных запятыми. На следующем этапе происходило преобразование полученных кодовых слов в индексы (токенизация). Заключительный этап – разбиение на сэмплы, в данном случае на пары, которые включают в себя последовательность токенов заданной длины и следующий за ней токен.

Нейронная сеть принимает на вход последовательности токенов, соответствующим кодовым словам, а на выходе выдает распределение вероятностей следующего элемента. Так как каждому выходу нейронной сети соответствует какой-то конкретный класс (токен), в качестве функции потерь выбрана категориальная перекрестная энтропия (categorical cross entropy).

Опираясь на то, что выводом нейронной сети являются подсчитанные для каждого класса вероятности того, что этот класс является следующим, было реализовано 3 способа получения следующего значения:

- Выбрать токен, вероятность, которого максимальна. Но в данном случае одинаковые последовательности будут всегда генерировать одинаковый результат.

- Выбрать из пяти наиболее вероятных токенов с равномерной вероятностью. Но бывают случаи, когда один из этих токенов имеет очень большую вероятность (порядка 90%), поэтому равномерная вероятность выбора не выглядит справедливой.

- Выбрать из пяти наиболее вероятных токенов, с вероятностью примерно соответствующей той, которую предсказала нейронная сеть.

Так как модель может только предсказывать следующий элемент последовательности, поданной на вход, для начала генерации необходимо подать некоторую последовательность токенов. Далее одним из описанных способов необходимо получить следующий токен. Полученным значением дополнить использованную последовательность. Из этой последовательности следует получить последовательность нужной длины (в данном случае взяв все элементы кроме первого) и использовать её в качестве нового входа сети. Данные действия необходимо повторять до тех пор, пока не накопится последовательность токенов необходимой длины.

Последний этап – сохранение мелодии. Полученные токены необходимо преобразовать в кодовые слова, которые соответствуют нотам и аккордам. Для сохранения полученной мелодии была использована библиотека music21, позволяющая записать список нот в виде файла в формате midi.

Для реализации нейронной сети и её обучения был использован язык программирования Python с библиотекой для глубокого обучения Keras, которая является надстройкой над фреймворком машинного обучения TensorFlow.

На рис. 3 показана архитектура применяемой сети, которая предназначена для работы с последовательностями длиной 100 и набором данных с 352 уникальными классами.

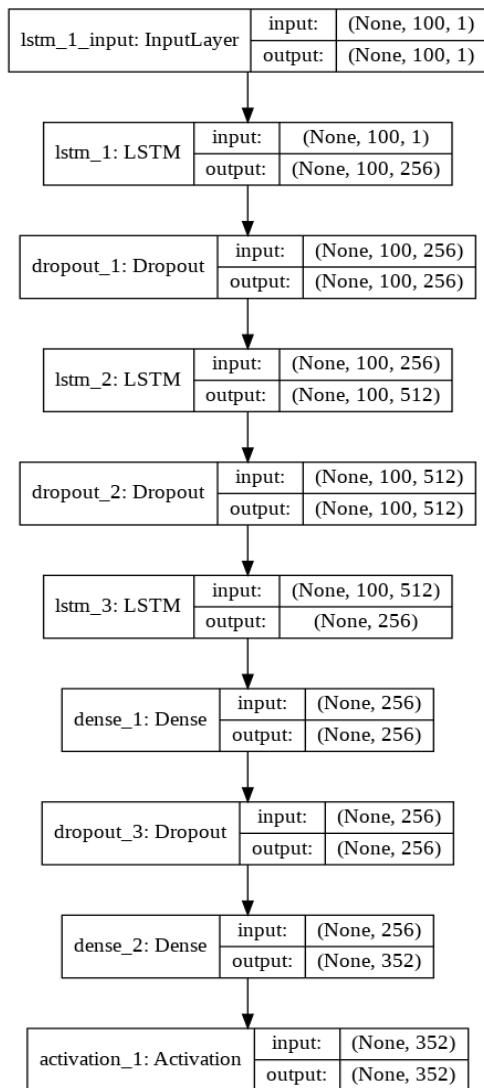


Рис. 3. Архитектура использованной сети

Заключение

В результате данного исследования была обучена нейронная сеть, способная создавать музыкальные записи произвольной длительности.

Полученные треки были схожи с теми, которые создают люди. Тем не менее, они все еще далеки от идеала. Основным их недостатком является излишняя монотонность. Возможно, использование более сложного способа представления мелодий, который учитывал бы длительность воспроизведения нот, позволило бы избежать указанного недочета за счет того, что мелодия с нотами разного размера звучит живее и динамичнее. Кроме того, модель не способна генерировать записи с нуля. Для решения этой проблемы необходимо использовать другую архитектуру сети, например, MuseGAN, которая была описана в [7].

Список литературы

1. Lewis, J. P. Creation by Refinement: A creativity paradigm for gradient descent learning networks / J. P. Lewis // Proceedings of the International Conference on Neural Networks. – San Diego, 1988. – Vol. 2. – P. 229-233.
2. Todd, P. M. A sequential network design for musical applications / P. M. Todd // Proceedings of the 1988 Connectionist Models Summer School. – San Mateo, 1988 – P. 76-84.
3. Olah, C. Understanding LSTM Networks / C. Olah [Электронный ресурс]. – Режим доступа : <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
4. Spezzatti, A. Neural Networks for Music Generation / A. Spezzatti [Электронный ресурс] : Towards Data Science. – Режим доступа : <https://towardsdatascience.com/neural-networks-for-music-generation-97c983b50204>
5. Спецификация MIDI [Электронный ресурс]. – Режим доступа : <https://www.midi.org/specifications>
6. Wibowo, H. A. Generate Piano Instrumental Music by Using Deep Learning / H. A. Wibowo [Электронный ресурс] : Towards Data Science. – Режим доступа : <https://towardsdatascience.com/generate-piano-instrumental-music-by-using-deep-learning-80ac35cddb2e>
7. MuseGAN : Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment / Hao-Wen Dong [et al.] // The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). – New Orleans, 2018 – P. 34-41. – Режим доступа : <https://arxiv.org/abs/1709.06298>

Разработка автоматизированного прибора измерения интенсивности проникающей радиации

Д. Н. Трофимов

Студент бакалавриата

С. А. Зув

Доцент

Введение

В 1895 году Рентгеном было открыто рентгеновское излучение, в 1896 Беккерель случайно обнаружил радиоактивность, а Томсон в 1897 обнаружил электрон. Эти три открытия показали, что в природе существуют ранее неизвестные излучения, способные отрывать электроны от атомов. По этой причине их назвали ионизирующими. В 1912 году Резерфорд опубликовал статью, в которой он сформулировал ядерную модель атома. После этого ионизирующие излучения стали называться ядерными [1].

В повседневной жизни человек подвергается воздействию различных источников ионизирующего излучения как естественного, так и искусственного (техногенного) происхождения.

- Естественный радиационный фон;
- техногенный фон от естественных радионуклидов;
- медицинское облучение за счет рентгено- и радиоизотопной диагностики;
- облучение, обусловленное работой предприятий атомной энергетики и промышленности и радиоактивным загрязнением окружающей среды в результате радиационных аварий и инцидентов.

Внимательнее остановимся на первых двух пунктах, потому что величина двух последующих постоянно контролируется большим числом специалистов.

1. Естественный радиационный фон

Часть облучения население Земли получает от естественных источников радиоактивного излучения. Большинство из них таковы, что избежать облучения от них совершенно невозможно. На протяжении всей истории существования Земли разные виды излучения падают на ее поверхность из космоса и поступают от радиоактивных веществ, находящихся в земной коре.

Человек подвергается облучению двумя путями. Радиоактивные вещества могут находиться вне организма и облучать его снаружи; в этом случае говорят о внешнем облучении. Или же они могут оказаться в воздухе, которым дышит человек, в пище или воде и попасть внутрь организма. Такой способ облучения называют внутренним.

Рассмотрим вначале некоторые данные о внешнем облучении от источников космического происхождения. Космические лучи в основном приходят к нам из глубин Вселенной, но некоторая их часть рождается на Солнце во время солнечных вспышек. Одни участки земной поверхности более подвержены их действию, чем другие. Северный и Южный полюсы получают больше радиации, чем экваториальные области, из-за наличия у Земли магнитного поля, отклоняющего заряженные частицы, из которых в основном и состоят космические лучи. Существеннее, однако, то, что уровень облучения растет с высотой, поскольку при этом над нами остается все меньше воздуха, играющего роль защитного экрана. Люди, живущие на уровне моря, получают в среднем из-за космических лучей эквивалентную дозу около 300 мкЗв/год; для людей же, живущих выше 2000 м над уровнем моря, эта величина в несколько раз больше.

Далее поговорим о земных радиоактивных источниках излучения. Разумеется, внешнее облучение неодинаково для разных мест планеты и зависит от концентрации радионуклидов в том или ином участке земной коры. В районах без выраженных геологических аномалий мощность дозы на открытой местности колеблется в диапазоне 40—60 нЗв/ч или около 350 мкЗв/год. Внутреннее облучение человека естественными радионуклидами происходит в результате накопления их в организме при поступлении с воздухом, питьевой водой и пищевыми продуктами. Средняя годовая эффективная доза внутреннего облучения жителей России составляет 380 мкЗв.

Доза радиации, получаемая при флюорографии, составляет 0,015 мЗв. При рентгенографии доза составляет 0,4 мЗв, а при рентгеноскопии – до 2,5 мЗв. При компьютерной томографии, для сравнения, получаемая доза может составлять 10 мЗв.

2. Техногенный фон от естественных радионуклидов

Под техногенным облучением обычно понимается облучение, обусловленное естественными радионуклидами, которые концентрируются в продуктах человеческой деятельности, например, строительных материалах, минеральных удобрениях, выбросах тепловых электростанций и др., т. е. техногенно-измененный естественный фон.

Основной вклад в дозу техногенного облучения дают строительные материалы, особенно содержащие повышенные концентрации калия, радия, тория — это пемза, некоторые марки бетона, литоидный туф, гранит, отходы переработки урана. Если в строительстве использованы материалы без повышенного содержания естественных радионуклидов, то независимо от вида материалов суммарная мощность дозы в помещениях примерно равна мощности дозы вне помещений. Средняя годовая эффективная доза внешнего облучения в помещениях только за счет излучения стройматериалов оценивается в 90 мкЗв.

Что касается внутреннего облучения за счет радона, то из стройматериалов и подстилающих пород в помещениях могут накапливаться концентрации, во много раз превышающие концентрации на открытой местности. В среднем в нормальных условиях это превышение 10—15-кратное. Еще один, как правило, менее значимый источник поступления этих радионуклидов в воздух помещений — это вода и природный газ. В воде поверхностных источников водоснабжения и газе их концентрации низки и практически не влияют на уровни облучения. Исключение может составить вода из артезианских скважин, в которой в зависимости от геологической структуры водоносных пород концентрация радона может быть достаточно высокой.

Конечно же нужно быть аккуратным с таким камнем как гранит. Не так редко встречаются новости о вагонах с ним с повышенным радиационным фоном. По этой причине его месторождения подлежат ежегодной сертификации, а поставки сопровождаются большим количеством документации, подтверждающей его безопасность. Большинство гранитов имеют уровень излучений в пределах 0,02 - 0,05 мкЗв/час. В какой-то мере природный камень соответствует области значений нормального наземного излучения. То есть плитка на улице в большинстве случаев безопасна.

Как мы видим, радиационный фон может сильно варьироваться, и поэтому знать о его состоянии вокруг себя может быть очень полезно, ведь если вы будете наблюдать повышенные значения в своем районе на протяжении долгого времени, можно будет задуматься о смене прописки. Поэтому я решил разработать комплексное решение, объединяющее в себе детектор ионизирующих излучений и средство для мониторинга среды [2].

3. Полупроводниковый детектор ионизирующих излучений

Для проекта был выбран полупроводниковый детектор – прибор для регистрации ионизирующих излучений, основным элементом

которого является кристалл полупроводника (чаще всего кремния или германия).

В кристалле за счет наличия областей с *n*- и *p*-проводимостями создается область, обедненная носителями (в ней электроны и дырки рекомбинируют). *p*-слой подсоединяется к отрицательному электроду, *n*-слой к положительному. Все носители оттягиваются от переходного слоя, диод заперт. Заряженная частица, проникая в детектор, создаёт дополнительные (неравновесные) электронно-дырочные пары, которые под действием электрического поля «рассасываются», перемещаясь к электродам прибора. В результате во внешней цепи полупроводникового детектора возникает электрический импульс, который далее регистрируется [3].

Самый известный и широко использующийся детектор — счетчик Гейгера-Мюллера. Полупроводниковый детектор, по сравнению с ним, имеет ряд преимуществ:

1. Энергия, затрачиваемая на образование пары электрон-дырка в полупроводниковом детекторе, меньше энергии образования пары электрон-ион в газах примерно в 10 раз. Следовательно, при полном торможении одной и той же частицы в импульсной камере и полупроводниковом детекторе амплитуда импульса в последнем примерно в 10 раз больше;

2. Разрешающее время полупроводниковых детекторов составляет примерно 10^{-8} секунд, в то время как длительность сигнала со счетчика Гейгера - 10^{-4} секунд — то есть время восстановления чувствительности полупроводникового детектора в 10000 раз меньше;

3. Счетчик Гейгера-Мюллера пригоден только для счета частиц, в то время как с помощью полупроводникового детектора можно определить и энергию частицы, и ее траекторию;

4. Высокая технологичность изготовления детекторов и сопутствующей электроники.

Но есть и некоторые недостатки (по сравнению с счетчиком Гейгера):

2. Многие полупроводниковые детекторы для правильной работы должны находиться при низких температурах (около 80К).

5. Несмотря на все недостатки, счетчики Гейгера-Мюллера не обладают 100%-ной эффективностью при регистрации ядерных частиц или γ -квантов. Это обусловлено тем, что частица, прошедшая через счетчик, может не создать даже одной пары ионов (либо ионы продиффундируют в нерабочую область). Тем не менее, эффективность счетчика к электронам может достигать 99% (и даже 99,9%).

Также присутствует ухудшение разрешающей способности при нагрузках $>10^4$ частиц в секунду и конечное время жизни полупроводникового детектора при высоких дозах облучения из-за накопления радиационных дефектов [4].

4. Разработка решения

Сначала был разработан детектор. Его схема представлена на рис. 1.

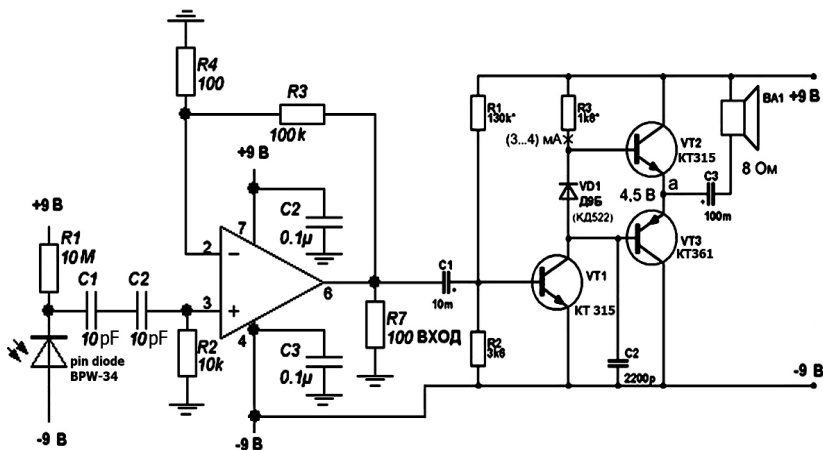


Рис. 1. Полупроводниковый детектор

Далее детектор, GPS-модуль и экран были подключены к плате Arduino UNO, с помощью которой создается пара: значение радиационного фона — координата.

На рис. 2:

1. pin-диод;
2. детектор;
3. GPS-модуль.

На микроконтроллере установлено программное обеспечение, формирующее документ специального вида, который можно загрузить на сайт для отображения полученной информации в виде меток на карте.

В дальнейшем предполагается разработать и изготовить на 3-D принтере пластмассовый корпус. Питание перевести на Li-ion аккумуляторные батареи типоразмера 18650 с DC-DC преобразователем на необходимые для функционирования устройства напряжениями.

Также предполагается снабдить устройство сетевым автономным питающим устройством для длительных измерений в помещениях.

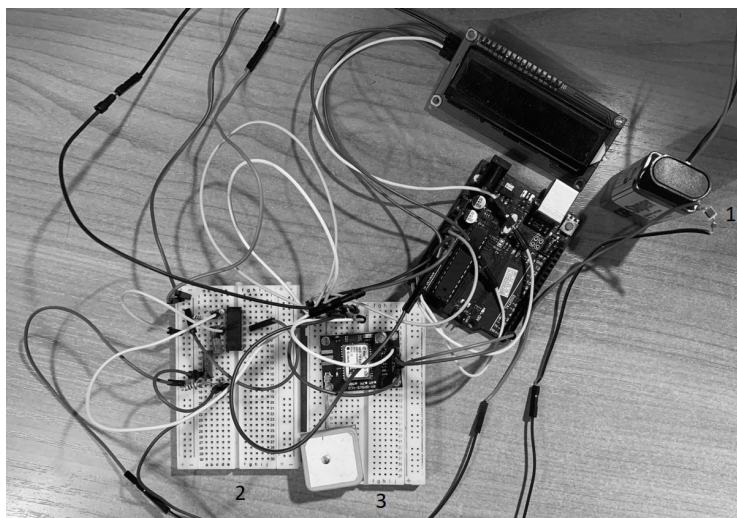


Рис. 2. Концепт прибора

Метка на сайте имеет цветовую индикацию, и при нажатии на нее появляется окно с полученными значениями. Исходный код проекта находится на GitHub [5]. На рис. 3 представлен макет сайта.

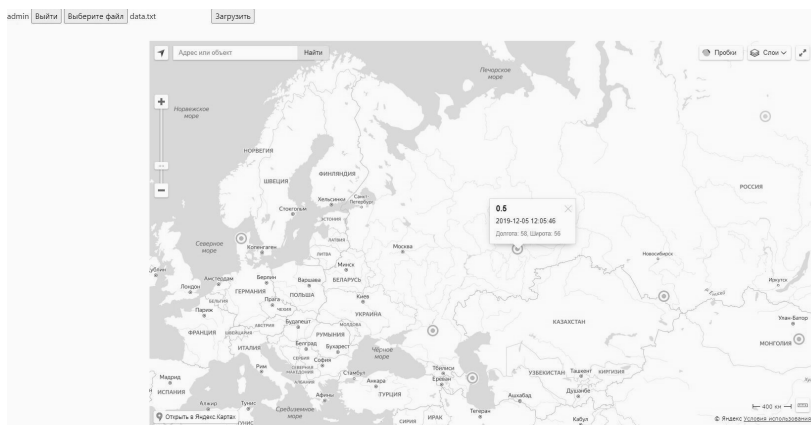


Рис. 3. Макет сайта

Заключение

Данная статья посвящена изучению воздействия ионизирующих излучений на человека. Рассмотрен полупроводниковый детектор, описан принцип его действия и преимущества по сравнению с другими датчиками. Разработано решение, включающее в себя прибор с низковольтным питанием и высокой точностью регистрации ионизирующих излучений и web-приложение для отображения информации о радиационном фоне. Планируется развитие данного датчика до полноценного прибора контроля за окружающей средой обитания человека путем добавления других датчиков, позволяющих отслеживать различные характеристики, такие как анализ химического состава воздуха, наличие и размер частичек пыли в нём, присутствие аллергенов.

Список литературы

1. Болоздыня, А. И. Детекторы ионизирующих частиц и излучений. Принципы применения : учебное пособие / А. И. Болоздыня, И. М. Ободовский. – Долгопрудный : Издательский Дом «Интеллект», 2012. – 208 с.
2. «Чернобыль в трех измерениях». Образовательная мультимедиа программа ИБРАЭ РАН, 2001-2006 [Электронный ресурс]. – Режим доступа : <http://www.ibrae.ac.ru/russian/chernobyl-3d/index.html>
3. Санкт-Петербургский государственный университет. Учебная лаборатория ядерной физики [Электронный ресурс]. – Режим доступа : <http://nuclab.spbu.ru/old/common/72.pdf>
4. Бараночников, М. Л. Приемники и детекторы излучений : справочник / М. Л. Бараночников. – М. : ДМК Пресс, 2012. – 640 с.
5. Nuclear_situation [Электронный ресурс]. – Режим доступа : https://github.com/Gena-gol/nuclear_situation

Возможность голосового набора программного кода – проблемы и подходы

А. Ю. Чиркин

Студент бакалавриата

Д. И. Соломатин

Старший преподаватель

Введение

В настоящее время технологии, использующие голосовой интерфейс, развиваются всё стремительнее: навигаторы, голосовые помощники, автоответчики, системы поиска, умные дома. В 2019 уже 20% всех поисковых запросов в Google на мобильных устройствах делаются с помощью голоса. Согласно прогнозу компании Ovum, к 2021 году общее число установленных голосовых помощников превысит 7,5 млрд. устройств [1]. Вместе с тем инструмента для набора программного кода голосом не существует.

1. Проблемы голосового набора кода

На первый взгляд может показаться, что набор кода голосом мало отличается от другой области применения голосовых интерфейсов, но это не так.

Во-первых, при наборе кода используются несуществующие в обычном языке слова – идентификаторы могут являться сокращениями, состоять из сокращений нескольких слов.

Во-вторых, кроме слов, код состоит из множества других символов: знаков препинания, скобок, кавычек, операторов, отступов и т. д. И, если просто выделить для этих символов специальные слова, то набор такого кода будет очень медленен и утомителен, так как доля этих специальных символов в коде велика.

Также, поскольку программист не ограничен в выборе идентификаторов то ошибки при распознавании будут случаться чаще, чем при работе с обычным языком. Поэтому необходимо предусмотреть возможность исправления ошибок при наборе.

Синтаксические конструкции языков программирования позволяют не проговаривать каждый символ. Скобки, кавычки всегда парные. Отступы в большинстве случаев можно выставить автоматически. Эти

конструкции также необходимо сделать удобными для набора голосом насколько это возможно.

2. Библиотека для распознавания речи

На сегодняшний день существует несколько библиотек для распознавания речи, среди которых была выбрана библиотека Pocketsphinx, т. к. она предоставляет возможность распознавания по пользовательскому словарю и работает офлайн.

Для распознавания речи библиотека требует 3 файла:

1. Акустическая модель. Строится для конкретного языка. Для английского языка она уже построена и доступна для использования.

2. Фонетический словарь. Фонетический словарь предоставляет системе отображение словарного запаса к последовательности фонем. Словарь должен содержать все интересующие слова, иначе библиотека не будет распознавать их.

3. Языковая модель. Используется для ограничения поиска слов. Определяет, какое слово может следовать за ранее использованным. Для распознавания слова оно должно присутствовать в языковой модели.

На вход библиотека принимает аудио-поток в виде последовательности бит. На выходе возвращает список так называемых гипотез – распознанных слов с коэффициентом, показывающим вероятность того, что именно это слово, по мнению Pocketsphinx, было произнесено (рис. 1).

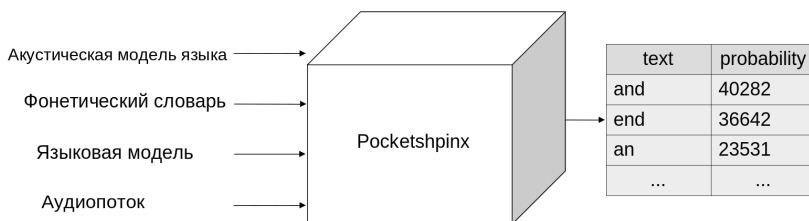


Рис. 1. Схема работы библиотеки

3. Повышение точности распознавания

Для повышения точности распознавания все возможные слова, которые требуется распознавать, были разбиты на словари:

1. часто используемые переменные;
2. встроенные функции;
3. операторы;
4. числа;

5. ключевые слова;
6. слова переключения контекста;
7. команды редактирования.

Прототип приложения создавался для языка Python, но идеи могут быть использованы для создания такого же инструмента и для других языков программирования.

Для получения списка часто используемых переменных был использован популярный репозиторий с алгоритмами (<https://github.com/keon/algorithms>). Оттуда с помощью скрипта были отобраны 80 наиболее часто встречающихся переменных. Самыми используемыми оказались переменные «i», «n», «node», «x», «value», «it», «key», «j», «current», «number», «root». Самыми непопулярными из отобранных оказались «sequence», «hash», «height», «line», «table», «second», «visited», «item», «total», «level».

В словарь операторов входят названия математические операции, знаки сравнения, всевозможные скобки, знаки препинания (всё в виде слов).

Словарь специальных команд содержит команды для переключения словарей, команды редактирования текста.

Эти фонетические словари генерировались при помощи онлайн инструмента «Sphinx Knowledge Base Tool» (<https://www.speech.cs.cmu.edu/tools/lmtool-new.html>).

Для повышения точности распознавания речи в соответствии с рекомендациями в документации библиотеки была проведена адаптация акустической модели [2]. Модель адаптируется к голосу говорящего и среде записи. Для адаптации требуется некоторое количество аудиозаписей с примерами. Были записаны более 200 команд из составленных словарей, а также написана транскрипция этих команд. Далее с помощью инструмента `sphinx_fe`, входящего в состав пакета CMUSphinx, была произведена адаптация акустической модели.

4. Общая схема работы

На рис. 2 представлена диаграмма деятельности приложения. Приложение работает в двух потоках. В одном потоке – графический интерфейс, в другом - модуль распознавания речи. Когда объект `recognizer` распознаёт фразу, то передаёт список гипотез объекту `parser`. `Parser` анализирует полученные данные и передаёт соответствующие команды интерфейсу. Например, напечатать слово или, наоборот, убрать последнее введённое слово.

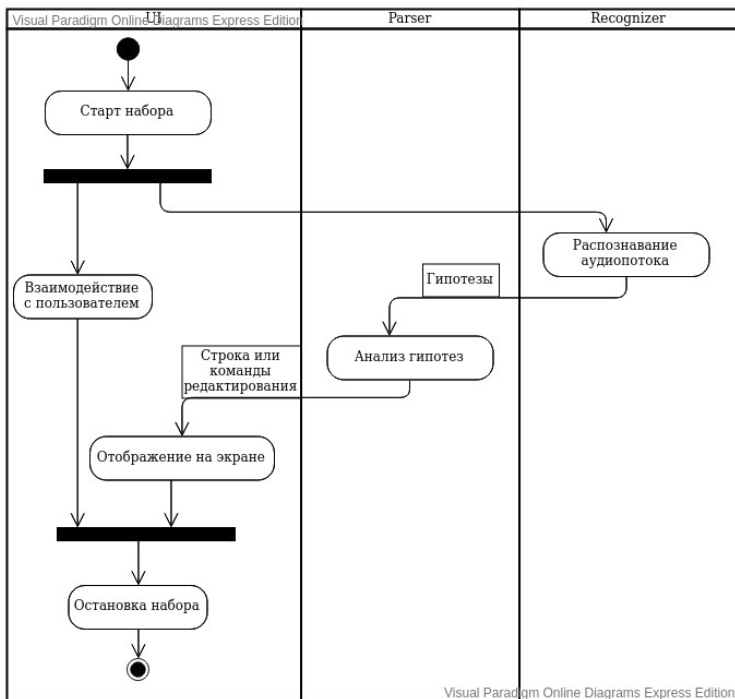


Рис. 2. Диаграмма деятельности

5. Анализ гипотез

У класса Parser есть контекст. Это переменная перечислимого типа. Она может принимать следующие значения:

1. instruction;
2. function;
3. select;
4. number;
5. variable;
6. string;
7. editing;
8. keyword;
9. sign.

В зависимости от значения этой переменной, распознавание ведётся по разным фонетическим словарям и языковым моделям.

Основной контекст – инструкция. Распознаются все слова из всех словарей. Наличие других контекстов позволяет использовать более

узкие словари, тем самым повысив вероятность правильного распознавания.

С помощью простейшей грамматики математические операторы, числа преобразуются в соответствующую форму. Также существуют некоторые слова для ускорения набора синтаксических конструкций. Например, слово «then» преобразуется в двоеточие, перевод строки и отступ.

Также наличие разных контекстов может пригодиться для написания зарезервированных слов. Например, слово «index» зарезервировано и означает печать квадратных скобок, чтобы объявить переменную «index» необходимо использовать более узкие словари, т.е. контекст с именами переменных.

Существуют специальные слова для смены контекста. Текущий контекст показывается пользователю. Например, если неправильно распознано имя переменной, то пользователь может отменить последнее действие (также специальной командой), переключить программу на контекст переменных и снова произнести имя переменной.

Некоторые примеры голосовых команд с пояснениями можно увидеть в таблице.

Таблица

Примеры голосовых команд

Команда	Пояснение
next line	Переход на следующую строку
indent	Отступ в 4 пробела
then	Набор двоеточия, переход на следующую строку и отступ
undo	Отмена последнего введённого слова
backspace	Удаление предыдущего символа
forward	Перемещение курсора вперёд на 1 символ
index	Набор открывающей и закрывающей квадратных скобок и размещение курсора между ними
tuple	Набор открывающей и закрывающей круглых скобок и размещение курсора между ними
select	Переключение парсера на контекст выбора из списка гипотез
variable	Переключение парсера на контекст переменных

6. Исправление ошибок

Даже при распознавании по ограниченному словарю случаются ошибки, поэтому был предложен способ для замены последней фразы.

При произнесении фразы самая лучшая гипотеза может оказаться ошибочной, но то, что было сказано на самом деле, наверняка будет где-то рядом.

На экране после каждой сказанной фразы появляется список слов-гипотез (рис. 3). Можно заменить последнее слово на какое-то из списка. Для этого нужно переключиться на контекст выбора из списка гипотез (произнеся «select») и назвать номер гипотезы. После чего последнее напечатанное слово заменится на выбранную гипотезу.

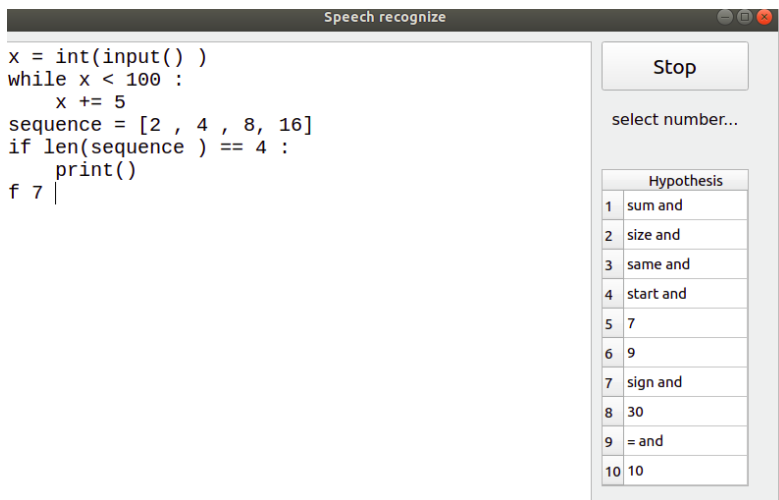


Рис. 3. Интерфейс программы

7. Результаты

Этот прототип создавался с целью исследовать различные подходы к решению задачи и, конечно, ещё далёк от настоящего инструмента.

Идеи о распознавании по нескольким более маленьким словарям, исправлении ошибок ввода с помощью предоставления списка гипотез для замены показали свою работоспособность. В ходе выполнения работы были выявлены следующие проблемы.

Точность распознавания даже на ограниченных словарях оставляет желать лучшего. Это может происходить из-за того, что при создании акустической модели английского языка использовались записи носителей английского языка, а не носители имеют большой акцент.

Предполагается, что для решения этой проблемы можно использовать акустические модели для русского языка. Тогда все вспомогательные команды можно будет говорить на русском, а для набора английских идентификаторов использовать транслитерацию.

Очевидно, что ограниченные наборы часто используемых идентификаторов недостаточны для нормального использования, поэтому необходимо динамически редактировать словари распознавания.

Было замечено, что проговаривать сокращения от имен идентификаторов не всегда удобно. Иногда удобнее говорить полное название (при этом набираться должно сокращение). Например, при произнесении «current» набираться должно «curr». Для этого нужно для каждого идентификатора предусмотреть варианты его сокращения и предлагать выбор из нескольких вариантов или, если идентификатор уже встречается в программе, использовать уже набранный вариант.

Для ускорения набора, когда вариантов следующих слов немного, можно сразу предлагать выбор из этих вариантов (например, после набора имени объекта можно предлагать идентификаторы его членов), аналогично подсказкам сред разработки. Это станет возможным с помощью подключения к API какой-либо среды разработки.

Заключение

Данная статья посвящена исследованию подходов к реализации программного кода. В результате выполнения работы создана основа для дальнейшего развития. Планируется продолжить данную работу в направлениях:

1. поддержка функций, классов;
2. повышение точности распознавания;
3. увеличение функциональности редактирования кода;
4. создание плагина для среды разработки.

Список литературы

1. Статья «Виртуальные цифровые помощники обгонят население мира к 2021 году» [Электронный ресурс]. – Режим доступа : <https://www.omdia.com/resources/product-content/virtual-digital-assistants-to-overtake-world-population-by-2021>
2. Документация библиотеки CMUSphinx [Электронный ресурс]. – Режим доступа : <https://cmusphinx.github.io/wiki/tutorial/>

Содержание

Акиндинова Т. В., Дрюченко М. А. Визуальное фенотипирование растений с использованием алгоритмов компьютерного зрения.....	3
Бобраков А. С., Гаршина В. В. Разработка программного модуля для интеллектуального анализа текстовых данных новостных интернет-ресурсов.....	11
Борисов А. Д., Соломатин Д. И., Климов К. В. Разработка персонализированного детектора ключевых точек лица.....	16
Будаева С. Ю., Копытин А. В. Решение задачи оптимального распределения ресурсов при планировании выпуска продукции в системе «1С:Предприятие 8.3».....	23
Верещагина М. С., Соломатин Д. И. Разработка прототипа приложения для автоматизации проверки решений задач по программированию в учебном процессе.....	30
Волгина В. П., Киселев Е. А. Приближенное построение двойственных систем с помощью рядов Фурье.....	37
Гаршин Т. С., Иванков А. Ю. Исследование игрового поведения участников on-line игр на основе онтологического подхода.....	42
Гостева Ю. В., Абрамов И. В. Повышение эффективности управления текущей деятельностью компании.....	48
Дремин М. Б., Головкин А. А. Разработка модулей сетевого ресурса и конфигурации сети для web-интерфейса SOHO-сервера.....	53
Дроботенко А. Д., Головкин А. А. Модель системы хранения документов об образовании с использованием технологии блокчейн.....	58
Еремин И. А., Чекмарев А. И. Система учёта успеваемости на факультете 2.0.....	64
Железной А. С., Дрюченко М. А. Применение методов машинного обучения для автоматического распознавания патологий растений.....	71
Захаров С. В., Абрамов И. В. Профилактика управления продажами коммерческой фирмы.....	79
Игнатов М. И., Головкин А. А. Алгоритм распределения канальных ресурсов в сетях когнитивного радио.....	84

Кищенко А. И., Каверина В. К. Моделирование распространения инфекционного заболевания COVID-19 на основе данных по Воронежской области с помощью SIR-модели.....	90
Косенко И. М., Тарасов В. С. Применение методов поиска геодезического расстояния для задачи понижения размерности 3D-модели с произвольной топологией.....	96
Крутских В. В., Лобода А. В. Использование символьных вычислений и компьютерных алгоритмов при интегрировании алгебр Ли.....	101
Кушнеренко В. К., Чекмарёв А. И. Система учета посещаемости на примере ФКН.....	107
Москаленко А. В., Быканов Н. С., Зуев С. А. Электронный датчик расхода воды.....	114
Никулин Р. А., Головкин А. А. Реализация анонимного ретранслятора IEEE 802.11.....	118
Олейник Е. И., Соломатин Д. И. Разработка прототипа приложения для построения блок-схем по коду алгоритмов.....	124
Палади А. С., Копытин А. В. Прогнозирование размера страховой медицинской выплаты пациентам методами машинного обучения.....	130
Пауков Н. В., Соломатин Д. И. Алгоритмы смешивания клипов для трёхмерной скелетной анимации.....	135
Подольский К. Д., Головин Я. Э., Копытина Е. А. Формирование предварительного коммерческого предложения ООО «ЧерноземАгромаш» средствами MS Excel.....	143
Попов В. Н., Киселев Е. А. Сравнительный анализ различных способов реализации метода канонических корреляций для цифровых сигналов.....	148
Саввина В. О., Киселев Е. А. Построение ортогональных систем целочисленных сдвигов.....	153
Семенихина Д. Е., Иванков А. Ю. Разработка программного модуля для решения задачи сегментации полупрозрачных объектов с потерей фокусировки на текстовых изображениях с его последующим восстановлением.....	158
Соколов В. А., Зуев С. А. Разработка радиоприёмника на базе микросхемы модели RDA5807.....	165
Стратиенко С. В., Головкин А. А. Разработка модуля "WWW" для web-интерфейса SOHO-сервера.....	170
Ступникова Е. П., Гаршина В. В. Система предобработки выбора контекстно-свободных грамматик для задач Text Mining.....	175

Теплякова М. А., Семёнов М. Е. Управление динамикой системы Малкуса–Лоренца посредством гистерезисных преобразователей.....	180
Толстов М. В., Соломатин Д. И. Применение нейронных сетей для генерации мелодий	185
Трофимов Д. Н., Зуев С. А. Разработка автоматизированного прибора измерения интенсивности проникающей радиации	192
Чиркин А. Ю., Соломатин Д. И. Возможность голосового набора программного кода – проблемы и подходы.....	199

Научное издание

**СБОРНИК СТУДЕНЧЕСКИХ НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ НАУК ВГУ**

В ы п у с к 14

**Часть 1
Научные работы студентов бакалавриата**

Под редакцией *Д. Н. Борисова*

Издано в авторской редакции

Подписано в печать 16.10.2020. Формат 60×84/16.

Усл. п. л. 12,1. Тираж 500 экз. Заказ 278

Издательский дом ВГУ

394018 Воронеж, пл. Ленина, 10

Отпечатано с готового оригинал-макета в типографии

Издательского дома ВГУ.

394018 Воронеж, ул. Пушкинская, 3