

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных наук

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

Выпуск 10

Под редакцией кандидата технических наук Д.Н. Борисова

Издательский дом
Воронежского государственного университета
2016

УДК 004.65 + 004.438.5

ББК Ч481(2)22

С23

Рекомендовано к печати

Ученым советом факультета компьютерных наук ВГУ

С23 **Сборник студенческих научных работ факультета компьютерных наук ВГУ / под ред. Д. Н. Борисова ; Воронежский государственный университет. – Выпуск 10. – Воронеж : Издательский дом ВГУ, 2016. – 240 с.**

ISBN 978–5–9273–2328–9

В сборник включены научные работы студентов факультета компьютерных наук ВГУ, выполненные в 2015–2016 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 978–5–9273–2328–9

© Воронежский государственный университет, 2016

© Оформление. Издательский дом ВГУ, 2016

ПРИЛОЖЕНИЕ ДЛЯ СОЗДАНИЯ СТЕРЕОИЗОБРАЖЕНИЙ

А.В. Аносова

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Человек видит мир трехмерным. Трехмерность достигается благодаря тому, что глаза человека видят «картинку» под разными углами. Этот эффект используется в стереоизображениях для создания иллюзии трехмерности. Ведь по своей сути стереоизображение – это лишь информация об образе, хотя и довольно точно имитирующая те зрительные ощущения, которые возникают при непосредственном восприятии окружающего мира.

Стереоизображение появилось чуть позже "обычной" двухмерной фотографии. Оно позволяет наблюдать трехмерные объекты без использования какого-либо специального оборудования.

В данной работе рассматривается задача создания таких стереоизображений, создание которых реализуется при помощи трехмерной сцены со всем набором объектов, расположенных на ней.

1. Принцип построения стереоизображений

Для построения стереоизображения необходима 3D-сцена. В контексте данной задачи под 3D-сценой можно понимать объект, позволяющий для любых двух координат X , Y определить третью координату Z (расстояние до точки от глаз наблюдателя).

3D-сцена может задаваться как непосредственно в виде 3D-модели сцены или же в виде карты глубины (в конечном итоге для создания карты глубины также необходима 3D-модель сцены).

Предположим, что 3D-сцена каким-либо образом задана. Рассмотрим, как по существующей 3D-сцене построить стереоизображение. Для создания стереоэффекта необходимо добиться, чтобы при просмотре стереоизображения [1] глаза человека смотрели на разные точки в плоскости картинки (лист бумаги, экран монитора). В этом случае точка пересечения линий взгляда будет восприниматься мозгом как точка на виртуальной стерео-поверхности, дальше плоскости реального изображения. На рис. 1. иллюстрируется данный принцип.

Добиться, чтобы глаза смотрели на разные точки, можно, если некоторые части изображения будут повторяться на каком-то расстоянии друг от друга (на рис. 1 это расстояние обозначено через l).

Для того чтобы получить расстояние h до виртуальной поверхности, расстояние l можно рассчитать по формуле:

$$l = \frac{h * L}{h + H}, \quad (1)$$

где h – глубина, L – расстояние между глаз, H – расстояние до изображения.



Рис. 1. Принцип построения стереоизображений

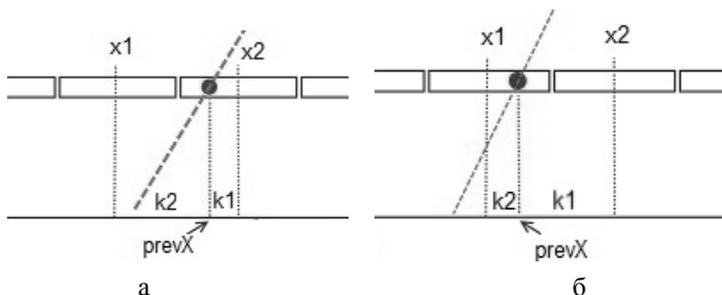
2. Борьба со ступенчатостью

Если при построении стереоизображения для нахождения точки, которую надо «повторить», округлять вычисленное значение l до целых (для копирования точек изображения строго «точка в точку»), то, в случае крупных точек, например, пиксели на экране монитора, появляется проблема ступенчатости получаемой виртуальной стерео-поверхности. Это объясняется тем, что дискретным (целым) значения l соответствуют дискретные значения h .

Для борьбы со ступенчатостью необходимо, чтобы глаза человека «фиксировались» не просто на разных точках стереоизображения, а «между точек», причем в заданном месте между центрами двух соседних точек (т. е. точки стерео-картинки рассматриваются как минимально возможные квадраты фиксированного цвета, из которых строится изображение). Такого эффекта можно добиться, если при вычислении цвета очередной точки стереоизображения, через которую должен проходить взгляд правого глаза, использовать пропорциональное

смешение цветов двух соседних точек, через которые должен проходить взгляд левого глаза. При этом соотношение цветов определяется расстоянием предполагаемой линии взгляда левого глаза от центра этих точек: чем ближе линия к центру точки, тем большая часть цвета берется из данной точки. В итоге, естественно, будет происходить постепенное сглаживание повторяющейся текстуры от левого края стерео картинки к правому. Но при этом, как показывает практика, стерео-поверхность будет восприниматься гладкой.

На рис. 2 и листинге ниже, продемонстрировано, как вычисляется цвет очередной точки при построении стереоизображения. Здесь $x1$ и $x2$ – центры точек, цвета которых используются для вычисления цвета очередной точки x ; $k1$ и $k2$ – соответствующие линейные коэффициенты, которые используются для покомпонентного вычисления цвета из цветов данных двух точек, причем $k1 + k2 = 1$ (расстояние между двумя соседними точками).



а – линия взгляда проходит ближе к центру точки $x2$, б – линия взгляда проходит ближе к центру точки $x1$

Рис. 2. Варианты прохождения линии взгляда левого глаза через соседние точки изображения (прямоугольники схематично изображают точки)

В разработанном приложении используется алгоритм построения стереоизображений, представленный на листинге 1 (приводится метод на языке C#).

В приведенном листинге zK – коэффициент глубины.

Листинг 1

Алгоритм построения стереоизображений

```
void makeStereoImage(ExtBitmap bmp, IZImage zImg,
                    double L, double H, double zK) {
    for (int y = 0; y < bmp.Height; y++) {
        for (int x = 0; x < bmp.Width; x++) {
            double h = zImg.getH(x, y);
```

```

double l = L * zK * h / (H + zK * h);
double prevX = x - l;

int x1 = (int) Math.Round(prevX);
if (x1 > prevX)
    x1 = x1 - 1;
int x2 = x1 + 1;
double k1 = x2 - prevX;
double k2 = 1 - k1;
if (x1 < 0)
    continue;

Color x1Color = bmp.Pixels[x1, y];
Color x2Color = bmp.Pixels[x2, y];
byte r = (byte) (x1Color.r * k1 + x2Color.r * k2);
byte g = (byte) (x1Color.g * k1 + x2Color.g * k2);
byte b = (byte) (x1Color.b * k1 + x2Color.b * k2);
bmp.Pixels[x, y] = Color.FromArgb(r, g, b);
    }
}
}

```

3. Интерфейс пользователя

Интерфейс приложения показан на рисунках ниже. Главное окно приложения содержит три закладки: «3D-сцена», «Карта глубины» и «Стереоиображение».

На закладке «3D-сцена» (рис. 3) пользователь может построить сцену из готовых 3D-моделей. Приложение включает в себя несколько десятков 3D-моделей, кроме того предусмотрена возможность загрузки новых моделей из файлов (формат ASE).

При переходе на закладку «Карта глубины» (рис. 4), пользователь видит карту глубины, построенную по заданной 3D-сцене.

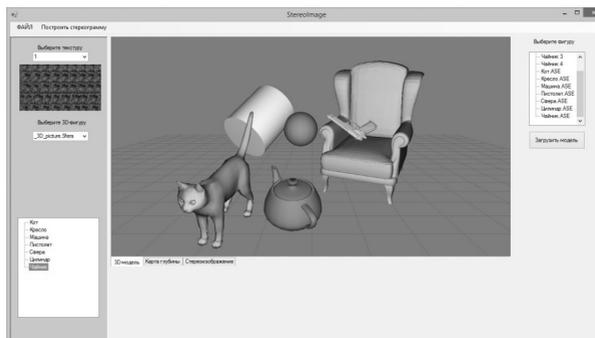


Рис. 3. 3D-сцена для расположения объектов

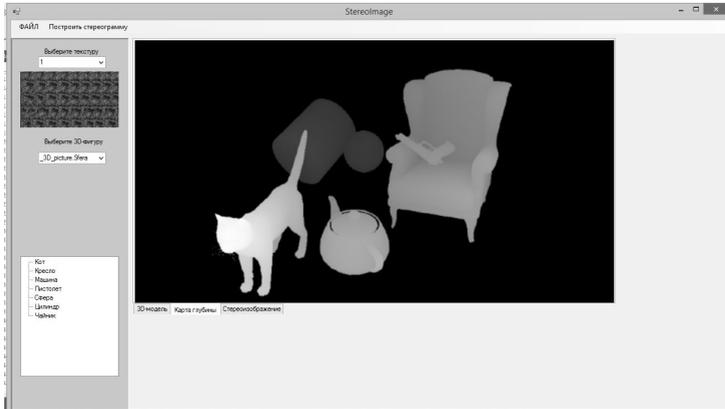


Рис. 4. Карта глубины, построенная по 3D-сцене

На закладке «Стереоблаженне» (рис. 5) происходит непосредственно построение стереоблаженне. Перед построением необходимо задать текстуру, которую можно выбрать из предлагаемого набора текстур или загрузить из файла.

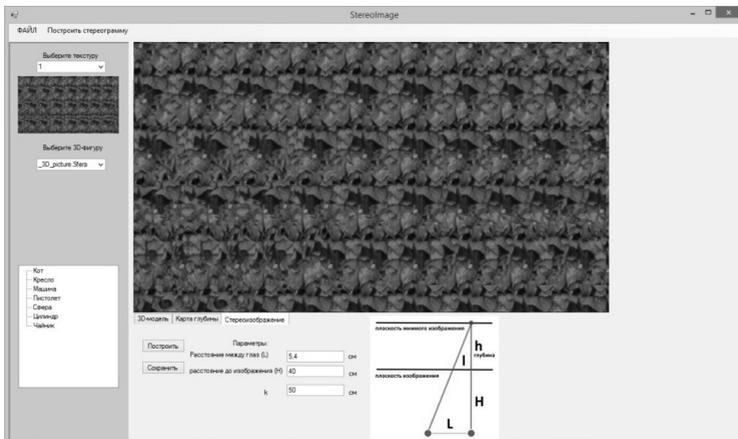


Рис. 5. Стереоблаженне, построенная по 3D-сцене (или карте глубины)

Заклучение

Доступные приложения не позволяют строить собственные 3D-сцены для построения стереограмм, а могут использовать экспортированные 3D-сцены в виде карт глубины. Создавать сцены можно только в полноценных 3D-редакторах, например, 3DMax, которые являются сложными для

освоения программами, к тому же большая часть из них являются платными и достаточно дорогими.

В отличие от существующих аналогов в разработанном приложении есть возможность создавать сцену с помощью добавления 3D-моделей, что является более гибким подходом по сравнению с построением стереоизображений по готовым картам глубины, для создания которых необходимо использовать сторонние приложения.

Список литературы

1. Стереоизображение – это просто [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/post/127681>

АНАЛИЗ РЕЗУЛЬТАТОВ ОПРОСОВ С ПОМОЩЬЮ ТАБЛИЦ СОПРЯЖЕННОСТИ, ФАКТОРИЗАЦИИ НЕОТРИЦАТЕЛЬНЫХ МАТРИЦ И ЛОГАРИФМИЧЕСКИ- ЛИНЕЙНЫХ МОДЕЛЕЙ

В.С. Бондаренко

Студент

А.П. Бубнов

Студент

П.С. Лысачев

Старший преподаватель

Введение

Опросы сегодня – явление очень распространенное. Их цель – знать мнение людей по определённой проблеме. Это наиболее эффективное средство анализа общественного мнения. Именно таким образом компании изучают спрос на свою продукцию, анализируют эффективность проведения рекламных кампаний. С помощью опроса можно определить целевую аудиторию и понять, чему отдают предпочтение разные слои населения.

В связи с развитием интернета и тем, что люди всё больше времени проводят в сети, популярность приобрели онлайн-опросы. Существует большое количество сервисов, предоставляющих возможность создания подобных опросов.

Самое основное в опросах это, конечно же, их результаты, которые анализируются и на их основе делаются выводы. Не всегда возможно проанализировать результаты просто посмотрев на них. В огромном

количестве цифр таблицы и в большом числе столбцов графика обнаружить закономерности будет невозможно или крайне затруднительно. В этом случае для поиска связей и более глубокого анализа данных приходят логарифмически-линейный анализ таблиц сопряженности [1], факторизация неотрицательных матриц и наглядное представление результатов их анализа с помощью тепловых карт (теплокарт) и графов.

1. Определение, виды и построение логлинейных моделей для таблиц сопряженности

Таблица сопряженности или факторная таблица – средство представления совместного распределения двух переменных, предназначенное для исследования связи между ними [1]. Она имеет вид представленный в таблице 1.

Таблица 1

Общий вид таблицы сопряженности

	Y_1	Y_2	...	Y_m	
X_1	F_{11}	F_{12}	...	F_{1m}	$F_{1.}$
X_2	F_{21}	F_{22}	...	F_{2m}	$F_{2.}$
X_3	F_{31}	F_{32}	...	F_{3m}	$F_{3.}$
...
X_n	F_{n1}	F_{n2}	...	F_{nm}	$F_{n.}$
	$F_{.1}$	$F_{.2}$...	$F_{.m}$	F

Пусть дана двухвходовая таблица сопряженности $I \times J$. Рассмотрим следующую модель.

Представим теоретические частоты в ячейках таблицы сопряженности в виде:

$$e_{ij} = e^{\lambda_0 + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB}} \quad (1)$$

или, в другой форме :

$$\ln e_{ij} = \lambda_0 + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB}, \quad (2)$$

где e_{ij} – теоретические (ожидаемые) частоты;

λ – неизвестные параметры, называемые:

λ_i^A – эффект i-ой категории признака А, $i = 1 \dots I$;

λ_j^B – эффект j-ой категории признака В, $j = 1 \dots J$;

λ_{ij}^{AB} – эффект взаимодействия двух признаков, $i = 1 \dots I, j = 1 \dots J$;

λ_0 – общий эффект.

Параметры λ называют вкладом, вносимыми различными эффектами в теоретическую частоту [2].

Параметры должны удовлетворять ограничениям [3]:

$$\sum_{ij} \lambda_{ij}^{AB} = \sum_i \lambda_i^A = \sum_j \lambda_j^B = 0, i = 1 \dots I, j = 1 \dots J \quad (3)$$

Подобные модели называются логлинейными (логарифмически-линейными) моделями таблицы сопряженности.

Для поиска связей между вопросами нужно рассчитать значения λ_{ij}^{AB} . Перед расчетом этих значений строится логарифмически-линейная модель, предполагающая независимость двух факторов. С помощью двух коэффициентов X^2 (4) и Y^2 (5) проверяется её адекватность.

$$X^2 = \sum_i \sum_j \frac{(f_{ij} - e_{ij})^2}{e_{ij}}, \quad (4)$$

$$Y^2 = 2 \sum_i \sum_j f_{ij} \ln \frac{f_{ij}}{e_{ij}}. \quad (5)$$

Если она не адекватна, тогда рассчитываются параметры λ_{ij}^{AB} по формуле (6).

$$\lambda_{ij}^{AB} = v_{ij} - v_{i.} - v_{.j} + v_{..}, i = 1 \dots I, j = 1 \dots J \quad (6)$$

где $v_{ij} = \ln e_{ij}$; $v_{..} = \sum_i \sum_j \frac{v_{ij}}{IJ}$ – общее среднее для логарифмов

вероятностей; $v_{i.} = \sum_j \frac{v_{ij}}{J}$ – среднее по всем логарифмам вероятностей, для

которых фактор A находится на уровне i ; $v_{.j} = \sum_i \frac{v_{ij}}{I}$ – среднее по всем логарифмам вероятностей, для которых фактор B находится на уровне j ; $i = 1 \dots I, j = 1 \dots J$.

Соотношение (6) показывает, что λ_{ij}^{AB} – это мера того, насколько совместное появление категорий A_i и B_j более (или менее) вероятно, чем можно было бы ожидать, если бы они были независимы.

Чем дальше значение λ_{ij}^{AB} отстоит от нуля, тем сильнее связь между категорией i переменной A и категорией j переменной B . На основе этого утверждения делается вывод о наличии связи и генерируется граф показанный на рис. 1. Значения λ_{ij}^{AB} могут быть отрицательными. Знак

"минус" указывает на то, что вследствие взаимодействия наблюдаемая частота в соответствующей ячейке таблицы сопряженности имеет меньшее значение, чем, если бы A и B были независимыми.

2. Факторизация неотрицательных матриц в анализе таблиц сопряженности.

Формально, алгоритм факторизации неотрицательных (Non-negative Matrix Factorization, далее NMF) матриц решает следующую проблему разложения матриц:

дана неотрицательная матрица V , найти матрицы W и H такие, что

$$V \approx W \times H. \quad (7)$$

В анализе данных опросов, NMF применяется для разложения таблицы сопряженности. Предположим, имеется таблица сопряженности размерностью $n \times m$, где n и m - количество вариантов ответов в первом и втором рассматриваемых вопросах. Эта матрица приблизительно может быть разложена на матрицу W размерностью $n \times r$, и H размерностью $r \times m$.

Обычно r выбирается такой, что он меньше n и m , таким образом матрицы W и H будут меньше, чем исходная матрица V .

Смысл аппроксимирования в (7) в том, что оно может быть переписано по столбцам $v \approx W \times h$, где v и h это соответствующие колонки матриц V и H . Другими словами, каждый линейный вектор v представлен линейной комбинацией столбцов из W , перемноженных на элементы h . Таким образом, можно сказать, что W содержит базис, который оптимизирован для линейной аппроксимации данных в V . Таким образом, несколько базисных векторов использованы для представления многих векторов данных. Достаточно точная аппроксимация может быть достигнута только если базисные векторы обнаруживают структуру, которая скрыта в данных.

Далее мы рассмотрим два алгоритма, базирующихся на итеративном обновлении W и H , потому что они легко реализуемы и их сходимость гарантирована.

Для начала, необходимо определить функции, которые будут количественно оценивать качество аппроксимации. Такие функции могут быть основаны на мере расстояния между двумя неотрицательными матрицами A и B . Одна из рабочих функций – квадрат Евклидова расстояния между A и B .

$$\|A - B\|^2 = \sum_{i,j} (A_{ij} - B_{ij})^2. \quad (8)$$

Другой вариант функции измерения расстояния выглядит так:

$$D(A\|B) = \sum_{i,j} (A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}). \quad (9)$$

Формально, эту функцию нельзя назвать расстоянием, потому что она не симметрична относительно A и B . Поэтому мы называем её «дивергенцией» A относительно B . В общем случае, она сводится к дивергенции К льбака - Лейблера, когда сумма элементов A равна сумме элементов в B .

В связи с (7) и (8), возникают две проблемы:

1. Минимизировать

$$\|V - WH\|^2, \quad (10)$$

с учетом того, что W и H неотрицательные.

2. Минимизировать

$$D(V\|WH), \quad (11)$$

с учетом того, что W и H неотрицательные.

Градиентный спуск — простейший путь решения данной проблемы. Недостатком этого метода является медленная сходимость. Ниже приведены правила обновления матриц W и H :

- для (10):

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T WH)_{a\mu}}, \quad W_{ia} \leftarrow W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}}. \quad (12)$$

- для (11):

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i \frac{W_{ia} V_{i\mu}}{(WH)_{i\mu}}}{\sum_k W_{ka}}, \quad W_{ia} \leftarrow W_{ia} \frac{\sum_i \frac{H_{a\mu} V_{i\mu}}{(WH)_{i\mu}}}{\sum_v H_{av}}. \quad (13)$$

В рамках данной работы, в приложении были реализованы оба алгоритма обновлений матриц (12), (13).

2. Цель эксперимента

Цель эксперимента в данной статье – поиск зависимостей между вопросами или опциями (вариантами ответов) с помощью факторизации неотрицательных матриц и логарифмически линейных моделей. Для данного эксперимента было разработано приложение на языке C# взаимодействующее с сервисом онлайн-опросов checkbox.com.

В качестве исходных данных были взяты два вопроса из тестового опроса [TestYourHobby](https://testyourhobby.com) запущенного на checkbox.com для демонстрации работы приложения.

Была построена таблица сопряженности для этих двух опросов. Далее была выдвинута гипотеза о независимости этих вопросов. Для проверки

гипотезы была построена логарифмически-линейная модель независимости, она показала ошибочность данной гипотезы. Затем, с помощью формул (4)-(6) рассчитываются значения λ_{ij}^{AB} . После оценки полученных значений был построен граф, отображающий найденные зависимости. Граф приведен на рис. 1.

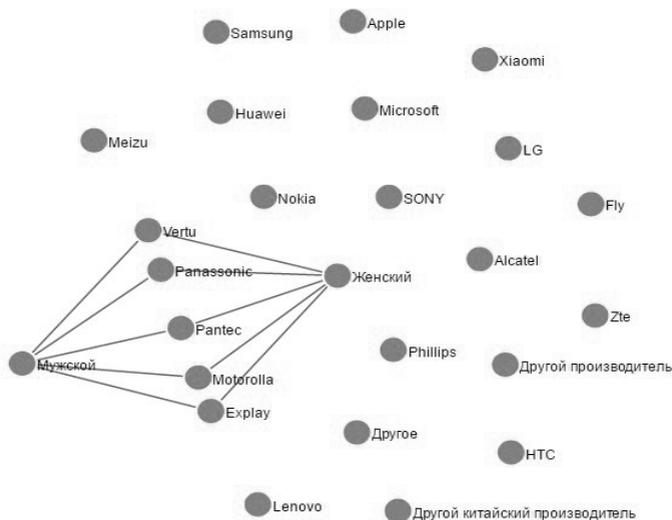


Рис. 1. Граф зависимостей

Так же, для двух вопросов с достаточно большим числом опций, (более 10) были построены теплокарты предпочтений респондентов. Для этого была построена таблица сопряженности, и для её разложения применены правила (13). Ранг разложения выбран равный 5. Наименования латентных переменных были даны самостоятельно на основе яркости выраженности опции на теплокарте.

Ниже представлен результат pmf разложения, исходная таблица сопряженности на рис. 2 и результирующие таблицы на рис. 3 и рис. 4, по которым можно увидеть основные связи опроса. Эффективность метода неочевидна при небольшом размере матрицы, однако, с увеличением размерности исходной таблицы, связи в ней выявить становится гораздо сложнее, чем по результирующим теплокартам. Стоит обратить внимание, что таблица на рис. 4 инвертирована для удобства чтения.

	Естественнонаучное	Радиотехническое	IT-сфера	Педагогическое	Экономическое	Юридическое	Историческое	Другое
Плавание	5.71	11.43	37.14	5.71	5.71	5.71	5.71	14.29
Бег	6	12	28	2	20	8	4	6
Единоборства	3.7	3.7	44.44	3.7	14.81	14.81	3.7	3.7
Бодибилдинг	0	13.33	23.33	3.33	6.67	6.67	3.33	33.33
Танцы	10.34	3.45	17.24	3.45	37.93	0	3.45	3.45
Велосипед	7.79	1.3	31.17	5.19	20.78	1.3	3.9	18.18
Сноуборд/лыжи	0	3.85	50	7.69	7.69	0	3.85	7.69
Страйкбол	6.67	0	26.67	6.67	0	13.33	6.67	30
Прослушивание музыки	6.67	1.67	28.33	4.17	17.5	6.67	5.83	13.33
Создание музыки	12	4	8	4	30	4	4	34
Вышивание	6.67	0	26.67	6.67	10	13.33	0	0
Собирание пазлов	8	4	32	4	24	4	0	8
Настольные игры	7.69	2.56	41.03	0	10.26	12.82	5.13	12.82
Шахматы, шашки...	3.03	0	42.42	0	3.03	6.06	3.03	15.15
Компьютерные игры	4.9	3.92	31.17	4.9	11.76	10.78	4.9	15.09
Гаджеты	6.25	6.25	37.5	0	15.63	9.38	3.13	12.5
Программирование	4.08	8.16	65.31	0	2.04	0	0	12.24
Изобретательство	0	7.14	42.86	7.14	7.14	7.14	0	7.14
Коллекционирование	0	0	23.53	11.76	17.63	23.53	5.88	5.88
Дом (ремонт, домашний уют)	2.78	5.56	25	0	13.22	2.78	2.78	11.11
Другое	9.09	4.55	27.27	4.55	0	9.09	0	31.82

Рис. 2. Исходная таблица сопряженности

Естественнонаучное	0	0.76	0	1	0
Радиотехническое	0.68	0.27	0	0.11	0
IT-сфера	1.13	0.3	2.57	0.68	0.03
Педагогическое	0	0	0.49	0	1.21
Экономическое	0	0	0	3.73	1.77
Юридическое	0	0	2.09	0	0
Историческое	0	0.27	0.28	0.18	0.47
Другое	0.11	0.99	0	0	1.03

Рис. 3. Результирующая матрица pmf разложения

Плавание	7.94	4.76	2.7	0.41	3.96
Бег	5.68	2.02	3.25	5.23	0.22
Единоборства	5.34	0.34	7.26	3.62	0.48
Бодибилдинг	5.64	5.51	2.29	0.09	1.07
Танцы	2.75	0.01	0	9.17	2.92
Велосипед	5.28	4.32	0.72	3.82	3.96
Сноуборд/лыжи	11.12	0	0	0	5.7
Страйкбол	0.05	6.91	3.84	0	0
Прослушивание музыки	3.11	3.57	3.71	3.67	2.49
Создание музыки	0	7.79	1.61	4.4	2.28
Вышивание	0.13	0	7.74	5.36	0
Собирание пазлов	5.33	1.73	1.86	5.36	1.72
Настольные игры	4.82	4.42	5.73	2.98	0
Шахматы, шашки...	6.35	4.35	2.94	0.76	0
Компьютерные игры	3.7	4.3	5.18	2.01	2.34
Гаджеты	6.03	3.91	3.6	4.06	0
Программирование	14.77	3.4	0	0.55	0
Изобретательство	8.85	0.13	2.97	0	4
Коллекционирование	0	0	9.82	0.34	7.32
Дом (ремонт, домашний уют)	4.35	3.16	1.07	3.14	0
Другое	3.87	10.07	4.77	0	0

Рис. 4. Результирующая матрица pmf разложения

Заключение

Данная статья посвящена обработке данных опросов с помощью построения таблицы сопряженности, логарифмически линейные модели и теплокарт на основе данных онлайн-опросов. В данной статье описан алгоритм анализа данных с помощью модели Гудмана [3]. Так же описан алгоритм факторизации неотрицательных данных, который впоследствии был применен к таблице сопряженности опроса для построения теплокарт, отображающих предпочтения пользователей. В итоге данного эксперимента получилась полностью рабочее приложение, позволяющее анализировать данные онлайн-опросов. Использование данного решения позволяет решать задачи анализа социологической информации, упрощает процесс изучения связей между переменными.

Список литературы

1. Аптон Г. Анализ таблиц сопряженности / Г. Аптон; под ред. Ю.А. Адлера. – М. : Финансы и статистика, 1982. – 145 с.
2. Goodman L.A. The Multivariate Analysis of Qualitative Data / L.A. Goodman. – N. Y. : Amer. Statist. Assoc., 1970. – 65 p.
3. Tutz G. Regression for Categorical Data / G. Tutz. – Cambridge : Cambridge University Press, 2011. – 561 p.
4. Daniel D. Lee Algorithms for Non-negative Matrix Factorization / D. Lee Daniel, H. Sebastian Seung. – Hanover : Now Publishers Inc., 2001. – 7 p.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ЗАДАЧ ПО ЯЗЫКУ SQL

И.А. Долгобородов

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Язык SQL в настоящий момент преподается в большинстве ВУЗов, занимающихся подготовкой специалистов в области информационных технологий. Проверка практических заданий по языку SQL занимает значительное время, т.к. одни и те же данные можно в SQL выбрать множеством различных способов. В «обычном» программировании давно «придуман» способ проверки алгоритмических задач с помощью тестов (решение проверяется на заранее подготовленном наборе

входных/выходных данных) [1]. Также существует множество систем для такого тестирования (а также проведения соревнований по спортивному программированию), как закрытых, например, Codeforces [2], так и открытых, например, EJudge [3]. Однако применение таких систем для заданий по языку SQL невозможно или неудобно.

В данной работе рассматривается задача разработки тестирующей системы для заданий по языку SQL.

1. Принцип проверки

Основой в языке SQL является запрос SELECT (запросы манипулирования данными достаточно простые и не вызывают затруднений). Как уже было сказано, написать SELECT-запрос можно различными способами, количество которых резко возрастает при увеличении сложности заданий. Естественно, проверить непосредственно правильность семантики запроса не представляется возможным, поэтому предлагается проверять результат выполнения запроса.

Для проверки результата выполнения SELECT-запроса необходимо иметь какой-то эталонный результат, который будет считаться правильным, при этом проверка решения будет, по сути, заключаться в сравнении двух наборов данных (таблиц): эталонного и полученного в результате выполнения запроса пользователя. Эталонный набор данных может быть получен в результате выполнения заведомо правильного запроса.

Таким образом, для каждого задания преподавателем должен быть написан правильный запрос (возвращающий правильный результат). Кроме этого формулировка заданий должна быть однозначной и достаточно подробной и, как минимум, включать в себя описание полей, которые необходимо выбрать, а также порядок сортировки данных. Естественно задания должны быть подобраны таким образом, чтобы результирующая выборка состояла из ограниченного числа строк (допустим, до 1000 строк).

Стоит отметить, что предложенный принцип проверки не является новым и используется, например, на сайте <http://sql-ex.ru> [4]. Однако данное решение является закрытым и, кроме того, намеренно не дает сравнивать результат запроса обучающегося и правильный результат.

2. Требования к тестирующей системе

При разработке рассматриваемой системы учитывались следующие требования:

- реализация в виде web-приложения;
- простота развертывания;
- возможность работы с различными СУБД;

- максимальная наглядность при демонстрации расхождений данных (правильного результата и результата запроса пользователя);
- устойчивость системы;
- хранение ответов пользователя для анализа преподавателем.

3. Реализация

Реализация тестирующей системы выполнена на языке PHP. Для работы с СУБД используется PHP-библиотека PDO, которая обеспечивает единый API для доступа к различным базам данных.

Особое внимание уделено устойчивости системы. В частности предусмотрено ограничение на время выполнения «тяжелых» запросов учащихся (написанных случайно или намеренно), а также на количество строк результата запросов.

Все данные системы хранятся на сервере в виде текстовых файлов, что обеспечивает максимальную простоту развертывания системы и добавления новых заданий. В частности задания добавляются в виде обычных текстовых файлов, в которых каждое задание представлено в виде текста условия и эталонного SELECT-запроса ниже. Таким образом, для подготовки заданий преподавателю даже доступ к данной системе не нужен.

Взаимодействие сервера и клиента (html-страница в браузере) происходит с помощью AJAX-запросов. При этом данные передаются в формате JSON, а их отображение на странице выполняется динамически с помощью JavaScript-библиотеки JQuery. В таблице 1 приведены предусмотренные типы таких запросов.

Таблица 1

Протокол взаимодействия клиента и сервера

Тип запроса	Параметры	Возвращаемое значение
getLabs	Нет	Список всех лабораторных работ, размещенных на сервере
getSettings	Нет	Настройки приложения
getStudents	Нет	Список всех зарегистрированных студентов
getStudentResults	name (строка) – имя студента	Информация о всех попытках выполнения заданий данным студентом

checkTask	lab (целое) – номер лабораторной; task (целое) – номер задания; query (строка) – SQL-запрос из textarea	Информация о результате проверки запроса пользователя в виде: – нескольких строк из выборки по запросу пользователя; – нескольких строк из выборки проверочного правильного запроса; – результат сравнения (номер несовпадающей строки и комментарий)
saveTask	lab (целое) – номер лабораторной; task (целое) – номер задания; query (строка) – SQL-запрос из textarea	

Ниже приводится пример запроса checkTask (листинг 1) и ответа на него (листинг 2).

Листинг 1

Вызов функции checkTask

```

...
var query = $("#query").value;
// lab и task уже выбраны в переменные
...
$.postJSON("ajax.php", {
    action: 'checkTask',
    lab: lab, task: task,
    query: query
}, function(data) {
    // отображение результата
    ...
})
);

```

JSON-ответ на вызов checkTask

```

{
  "message": "Несовпадает 3-я строка",
  "first_wrong_row_index": "3",
  "query_rows": {
    "columns": ["customer_id", "cust_name", "sum2000"],
    "rows_count": "319",
    "rows": {
      "1": ["101", "Welles Constantin", "82445.60"],
      "2": ["102", "Pacino Harrison", "null"],
      "3": ["103", "Taylor Manisha", "10000.00"],
      "4": ["104", "Sutherland Harrison", "46257.00"],
      "5": ["105", "MacGraw Matthias", "37299.80"]
    }
  },
  "right_rows": {
    "columns": ["customer_id", "cust_name", "orders_sum"],
    "rows_count": "319",
    "rows": {
      "1": ["101", "Welles Constantin", "82445.60"],
      "2": ["102", "Pacino Harrison", "nul"],
      "3": ["103", "Taylor Manisha", "null"],
      ...
    }
  }
}

```

4. Интерфейс пользователя

Как уже было сказано, приложение имеет односторонний интерфейс, данные обновляются динамически. Пожалуй, отдельно стоит рассмотреть только отображение расхождений результата запроса обучающегося с результатом правильно запроса. На рисунке показано отображение результата запроса `checkTask` для данных, приведенных на листинге 2.

В разработанном приложении предусмотрено два режима доступа: обучающийся и преподаватель. В режиме «обучающий» пользователь только может тестировать свои запросы на правильность и сохранять их, когда сочтет нужным (этот интерфейс как раз и показан на рисунке). В режиме «преподаватель», пользователю доступны для просмотра и комментирования сохраненные запросы учащихся, а также сводные данные по результатам обучающихся.

localhost localhost/test/index2.php

ivan lab1

Задание №7

Выбрать информацию обо всех заказчиках и сумме их заказов за 2000. Вывести поля: код заказчика, имя заказчика (фамилия + имя через пробел) за 2000 год. Упорядочить по коду заказчика.

```

1 select c.customer_id,
2       concat(c.cust_last_name, ' ', c.cust_first_name) as cust_name,
3       /* имитация ошибки */
4       if(c.customer_id=103, 10000, sum(o.order_total)) as sum2000
5 from   customers c
6       left join orders o
7       on o.customer_id=c.customer_id AND
8          '2000-01-01'<o.order_date and o.order_date<'2001-01-01'
9 group by c.customer_id
10      order by c.customer_id

```

не совпадает 3-я строка

Ответ на ваш запрос (всего строк:319)

№ строки	customer_id	cust_name	sum2000
1	101	Welles Constantin	82445.60
2	102	Pacino Harrison	null
3	103	Taylor Manisha	10000.00
4	104	Sutherland Harrison	46257.00
5	105	MacGraw Matthias	37299.80
...

Правильный ответ (всего строк:319)

№ строки	customer_id	customer_name	orders_sum
1	101	Welles Constantin	82445.60
2	102	Pacino Harrison	null
3	103	Taylor Manisha	null
4	104	Sutherland Harrison	46257.00

Расхождение результата запроса пользователя и правильного «ответа»

Рисунок. Интерфейс пользователя

На данном рисунке можно видеть, что строки из запроса пользователя, которые совпадают с правильным «ответом», отображаются зеленым цветом, первая строка с расхождением отображается красным цветом, а соответствующая ей строка из правильного «ответа» выделена.

Закключение

Естественно, принцип тестирования, который заложен в созданной тестирующей системе, не может гарантировать, что запрос учащегося написан правильно, даже если выбранный набор данных совпадает с правильным. Выявление такого рода логических ошибок по-прежнему лежит на преподавателе, однако применение данной системы гарантированно облегчает труд последнего, т. к. ему не приходится по несколько раз проверять неправильно написанные запросы учащихся.

Список литературы

1. Андреева Е.В. Принципы проверки учебных и олимпиадных задач по информатике / Е.В. Андреева [Электронный ресурс]. – Режим доступа : <http://информатика.1сентября.рф/2001/9/art/35.html>
2. Codeforces. Соревнования по программированию / (с) Copyright 2010-2016 Мирзаянов М. [Электронный ресурс]. – Режим доступа : <http://codeforces.com/>
3. Чернов А. В. Система тестирования ejudge / А. В. Чернов // Вторая конференция "Свободное программное обеспечение в высшей школе". – Переславль-Залесский, 2007 [Электронный ресурс]. – Режим доступа : <http://heap.altlinux.org/pereslavl2007/chernov/abstract.html>
4. Упражнения по SQL [Электронный ресурс]. – Режим доступа : <http://sql-ex.ru/>

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ МИКРОКЛИМАТОМ В ЗАКРЫТЫХ ПОМЕЩЕНИЯХ

Д.А. Душкин
Студент
С.А. Зуев
Ассистент

Введение

Контроль микроклимата в закрытых помещениях является важной, а в некоторых случаях, неотъемлемой частью производственного процесса. В данной статье будет рассматриваться управление микроклиматом в овощехранилище.

Технология хранения сельхозпродукции включает в себя процессы регулировки температуры и влажности в овощехранилище с целью предохранения продукции от переохлаждения, перегрева и обезвоживания.

Основные параметры микроклимата в хранилищах — температура и относительная влажность воздуха в массе хранимого продукта.

Процессы управления микроклиматом овощехранилищ рассмотрим на примере хранения картофеля [1].

В технологическом процессе хранения картофеля можно выделить три периода: лечебный, охлаждения и хранения.

Лечебный период необходим для быстрого заживления механических повреждений картофеля.

С этой целью в межклубневом пространстве насыпи необходимо поддерживать температуру на уровне 14...18 °С и высокую относительную влажность воздуха (более 90 %) с минимальным воздухообменом.

При температуре картофеля выше 18 °С должна включаться система активного вентилирования и подавать воздух температурой на 3...4 °С ниже температуры массы хранимого продукта.

Если в закрома хранилища заложен большой картофель (пораженный фитофторой, нематодой и т. п.), то лечебный период проводится при температуре 8...10 °С с последующим охлаждением до 1...2 °С.

При закладке мокрого картофеля его немедленно подсушивают усиленным активным вентилированием при относительной влажности воздуха не более 80 %.

Период охлаждения наступает после двухнедельного лечебного периода, температуру хранимого картофеля постепенно снижают до 2...4 °С. Охлаждают клубни медленно: на 0,5...0,6 °С в сутки при максимальной влажности воздуха до 100 %. Период охлаждения длится 20-25 суток.

Период хранения (основной) начинается при температуре картофеля в насыпи 3...4 °С. Вентиляционные установки включаются при температуре в насыпи 4 °С и более.

Для реализации систем автоматического регулирования климатом в овощехранилищах по алгоритмам любой сложности, необходимо наличие управляющего устройства (микроконтроллера) и сенсоров (датчиков), чувствительных к изменению регулируемых параметров. В рассматриваемом случае — датчиков температуры и влажности. Для справки выводится и значение атмосферного давления. Кроме того, система должна иметь возможность регулировать необходимые параметры измеряемых величин, но эта часть системы (регулирующая) выходит за рамки настоящей статьи и в ней не рассматривается. Перейдём к обзору датчиков и микрокомпьютеров.

1. Оборудование

1.1 Датчики температуры

Терморезистивные термодатчики — основаны на принципе изменения электрического сопротивления (полупроводника или проводника) при изменении температуры [2].

Эти датчики состоят из металла, чаще всего платины. В принципе, любой металл изменяет свое сопротивление при воздействии температуры, но используют платину, так как она обладает долговременной стабильностью, прочностью и воспроизводимостью характеристик. Для измерений температур более 600 °С могут использоваться также вольфрам.

Минусом этих датчиков является высокая стоимость и нелинейность характеристик.

Кремневые резистивные датчики. Преимущества этих датчиков — хорошая линейность и высокая долговременная стабильность.

Термисторы. Эти датчики изготавливаются из металл-оксидных соединений. Датчики измеряют только абсолютную температуру.

Полупроводниковые датчики регистрируют изменение характеристик р-п перехода под влиянием температуры. В качестве термодатчиков могут быть использованы любые диоды или биполярные транзисторы. Пропорциональная зависимость напряжения на транзисторах от абсолютной температуры (в Кельвинах) дает возможность реализовать довольно точный датчик.

Термоэлектрические преобразователи — иначе, термопары. Они действуют по принципу термоэлектрического эффекта. Так, один конец термопары (рабочий) погружен в среду, а другой (свободный) — нет. Таким образом, термопары это относительные датчики и выходное напряжение будет зависеть от разности температур двух частей. И почти не будет зависеть от абсолютных их значений. Диапазон измеряемых с их помощью температур, от -200 до 2200 градусов, и напрямую зависит от используемых в них материалов.

Пирометры — бесконтактные датчики, регистрирующие излучение, исходящее от нагретых тел. Основным достоинством пирометров (в отличие от предыдущих температурных датчиков) является отсутствие необходимости помещать датчик непосредственно в контролируемую среду.

Различают три вида пирометров:

Флуоресцентные. При измерении температуры посредством флуоресцентных датчиков на поверхность объекта, температуру которого необходимо измерить, наносят фосфорные компоненты. Затем объект подвергают воздействию ультрафиолетового импульсного излучения, в результате которого возникает после излучения флуоресцентного слоя, свойства которого зависят от температуры. Это излучение детектируется и анализируется.

Интерферометрические. Интерферометрические датчики температуры основаны на сравнении двух лучей — контрольного и пропущенного через среду, параметры которой меняются в зависимости от температуры. Чувствительным элементом этого типа датчиков чаще всего выступает тонкий кремниевый слой, на коэффициент преломления которого, а, соответственно, и на длину пути луча, влияет температура.

Датчики на основе растворов, меняющих цвет при температурном воздействии. В этом типе датчиков-пирометров применяется хлорид

кобальта, раствор которого имеет тепловую связь с объектом, температуру которого необходимо измерить. Коэффициент поглощения, которого зависит от температуры. При изменении температуры меняется величина прошедшего через раствор света.

1.2. Датчики влажности

Емкостные датчики влажности широко используются в промышленном оборудовании.

Такие датчики конструктивно состоят из подложки, на которой расположен тонкопленочный полимерный или металлооксидный между двумя проводящими электродами. Инкрементальные изменения в диэлектрической константе емкостного датчика влажности практически прямо пропорциональны относительной влажности окружающего воздуха. При колебании влажности на 1% емкость изменяется на 0.2-0.5 пФ, а при 50% влажности (25°C) колебания могут достигать от 100 до 500 пФ. [3]

Емкостные датчики влажности характеризуются низким температурным коэффициентом, возможностью работы на высоких температурах (вплоть до 200°C), возможностью полного восстановления при попадании конденсата и умеренной стойкостью к химическим испарениям.

Типичный класс точности емкостных датчиков составляет $\pm 2\%$ в диапазоне от 5 до 95% относительной влажности при калибровке по двум точкам.

Резистивные датчики влажности фиксируют изменения электрического сопротивления гигроскопической среды (например, проводящего полимера, соли или обработанной подложки). Их сопротивление оказывается обратно пропорциональным влажности. Диапазон сопротивлений типичного резистивного элемента колеблется от 1 кОм до 100 МОм. Точность составляет $\pm 2\%$.

1.3. Микрокомпьютеры

Современный рынок микрокомпьютеров (вычислительных устройств размером с банковскую карточку) насчитывает более полутора сотен штук, которые различаются быстродействием, объемом оперативной и перепрограммируемой памяти, развитым интерфейсом ввода-вывода, потребляемой мощностью, ценой, развитой системой разработки программ и т. д. Остановимся вкратце на некоторых общедоступных микрокомпьютерах.

HummingBoard представляет собой семейство микрокомпьютеров на базе процессоров архитектуры ARM Cortex, разрабатываемое компанией SolidRun. На данный момент состоит из четырех моделей, каждая из которых работает под управлением ОС (Linux/Android) и обладает

избыточными для рассматриваемой в данной статье задачи объемом памяти (0,5-4 ГБ ОЗУ) и быстродействием (SOM на базе ARM A9, 1-4 ядра, 1ГГц). [4]

Цена варьируется от 70 до 252 долларов США в зависимости от комплектации.

Микрокомпьютер *Black Swift* разработан компанией "Умная электроника". Чипсет – Qualcomm Atheros AR9331. 16 МБ постоянной памяти, 64 МБ оперативной памяти. Интерфейсы ввода-вывода: 1×USB 2.0, 26×GPIO, 1×PS/SPDIF, 2×Fast Ethernet, 1×SPI, 1×I²C, 1×16550 UART. Особенностью являются малые габариты: 25×35×4 мм. Работает под управлением операционной системы OpenWRT 14.07 [5].

Стоимость: 1500-2500 рублей.

BeagleBone Black — микрокомпьютер, разработанный компанией Texas Instruments. BeagleBone Black построен на базе процессора Sitara XAM3359AZCZ100 семейства ARM Cortex A8, который работает на частоте 1 ГГц, 512 МБ оперативной памяти DDR3L на шине с частотой 800 МГц и 4 ГБ флэш-памяти eMMC [6].

На плате расположены следующие разъемы для подключения внешних устройств: micro-HDMI, mini-USB, RJ45-разъём, 2 колодки по 46 выводов для подключения электронных модулей и компонентов. Из 92 выводов на колодках 65 могут быть использованы для цифрового ввода и вывода общего назначения (GPIO). При этом некоторые из них предоставляют дополнительные возможности: 8 каналов ШИМ на 4 независимых таймерах; 7 аналоговых входов, подключённых к 12-битному АЦП (4096 градаций); 4 последовательных интерфейса UART и 1 дополнительный UART только с линией передачи (TX); 2 шины TWI/I²C; 2 шины SPI; 25 портов ввода-вывода реального времени. Порты ввода-вывода реального времени (PRU) подключены к встроенному микроконтроллеру на 200 МГц.

Стоимость: 49-89 долларов США в зависимости от комплектации.

Raspberry Pi — микрокомпьютер, разработанный Raspberry Pi Foundation. Первый прототип этого устройства появился в 2006 году, и предназначался для обучения школьников азам компьютерных премудростей. Рассмотрим характеристики Raspberry Pi Zero, так как это наиболее дешёвый и подходящий для рассматриваемой задачи вариант Raspberry Pi. Процессор Broadcom BCM2835 (ядро ARM11, 1ГГц), 512Мб ОЗУ (LPDDR2 SDRAM). Присутствует слот для MicroSD карт, разъемы mini-HDMI и micro-USB, также стоит отметить наличие 40 выводов общего назначения (GPIO) и композитный видеовыход. Работает под управлением Linux. Заявленная цена — пять долларов США [7].

Arduino Uno построен на базу микроконтроллера ATmega328P, имеет 32 КБ флэш-памяти и 2 КБ ОЗУ.

Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP [8].

Операционной системы не имеет, программируется посредством ПО Arduino. Также имеется большое число библиотек для работы с дисплеями, датчиками, сервоприводами, Ethernet, Wi-Fi, GSM сетями и так далее, что существенно упрощает программирование.

Средняя стоимость платы Arduino Uno в интернет магазинах Китая на момент написания статьи составляет около трех-четырёх долларов.

Учитывая соотношение стоимости и удобства программирования микрокомпьютера, останавливаем свой выбор на Arduino Uno.

2. Решение задачи

Целью является построение системы, которая:

- Считывает показания с четырех датчиков (двух датчиков температуры, датчика влажности, датчика давления);
- Выводит на дисплей температуру в помещении, температуру вне помещения, влажность в помещении, атмосферное давление;
- Подает управляющие сигналы регулирующей части в зависимости от заданных условий.

Для решения задачи используется микроконтроллер Arduino Uno для управления системой и LCD-дисплей LCD1602 I2C для вывода информации с датчиков и состояния регулирующей части системы.

Датчик, измеряющий температуру в помещении подключен к аналоговому входу A0, измеряющий температуру вне помещения — к A1, измеряющий давление — к A2, измеряющий влажность — к A3.

Управляющие сигналы организованы следующим образом:

- для управления увеличением температуры подается высокое напряжение (+5 В) на цифровой выход D8
- для уменьшения температуры — на выход D9;
- для увеличения влажности — на выход D10;
- для уменьшения влажности — на выход D11.

Важно отметить, что на выходы D8 и D9 (или D10 и D11) не может одновременно подаваться высокое напряжение.

В программной части используются стандартная библиотека Wire и библиотека NewLiquidCrystal [9].

Для преобразования значений напряжения на входах (0...+5 В) в значения требуемых физических величин используются стандартные функции библиотек: `get_temp_from_voltage`, `get_humidity_from_voltage`,

get_pressure_from_voltage; для вывода данных на дисплей используются функции print_temperature, print_humidity, print_pressure.

Управляющие сигналы устанавливаются следующим образом:

– Если температура в помещении меньше LOW_TEMP, то на D8 подается высокое напряжение, на D9 — низкое. Если температура в интервале (LOW_TEMP, HIGH_TEMP), то на D8 и D9 подается низкое напряжение. Если температура больше HIGH_TEMP, то на D8 подается низкое напряжение, на D9 — высокое.

– Если влажность в помещении меньше LOW_HUMIDITY, то на D10 подается высокое напряжение, на D11 — низкое. Если влажность в интервале (LOW_HUMIDITY, HIGH_HUMIDITY), то на D10 и D11 подается низкое напряжение. Если влажность больше HIGH_HUMIDITY, то на D10 подается низкое напряжение, на D11 — высокое.

Полный текст программы доступен по адресу [10].

Заключение

Данная статья посвящена созданию замкнутой системы управления микроклиматом в овощехранилище. Описано создание программно-аппаратного комплекса на базе микрокомпьютера Arduino Uno, который успешно решает задачи измерения параметров микроклимата и управления регулирующей частью системы управления микроклиматом. Описанный комплекс характеризуется низкой стоимостью, легкостью в создании и эксплуатации, что подчёркивает целесообразность выбора платформы Arduino.

Источники

1. Сборник статей и лекций [Электронный ресурс]. — Режим доступа : <http://lektcii.com/1-90746.html>
2. Описание датчиков температуры [Электронный ресурс]. — Режим доступа : <http://www.devicesearch.ru/article/datchiki-temperature>
3. Описание датчиков влажности [Электронный ресурс]. — Режим доступа : <http://www.sensorica.ru/docs/art3.shtml>
4. Официальный сайт производителя HummingBoard [Электронный ресурс]. — Режим доступа : <https://www.solid-run.com/freescale-imx6-family/hummingboard/>
5. Официальный сайт производителя Black Swift [Электронный ресурс]. — Режим доступа : <http://www.black-swift.ru/>
6. Официальный сайт BeagleBoard [Электронный ресурс]. — Режим доступа : <http://beagleboard.org/>
7. Официальный сайт производителя Raspberry Pi [Электронный ресурс]. — Режим доступа : <https://www.raspberrypi.org/>

8. Официальный сайт Arduino [Электронный ресурс]. — Режим доступа : <https://www.arduino.cc>
9. Описание и файлы библиотеки NewLiquidCrystal [Электронный ресурс]. — Режим доступа : <https://bitbucket.org/fmalpartida/newliquidcrystal/wiki/Home>
10. Open source community [Электронный ресурс]. — Режим доступа : https://github.com/brandstein/microclimate-controller/blob/master/microclimateController_I2C_LCD_ino

ФРЕЙМВОРК ДЛЯ ПОДГОТОВКИ СОРЕВНОВАНИЙ ПО ПРОГРАММИРОВАНИЮ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

М.В. Землянухин

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Соревнования по программированию с каждым годом становятся все популярнее. И это связано не только с ростом популярности программирования в целом, но и с повышением качества организации соревнований, развитием олимпийского движения, увеличением заинтересованности работодателей в программистах, способных решать сложные и нестандартные задачи.

Существуют различные виды соревнований по программированию, начиная от классической схемы, в которой команды из трех человек, в распоряжении которых находится один компьютер, за ограниченное время решают определенный набор задач разного уровня сложности, до многочасовых соревнований, в которых перед участниками стоит задача разработать прототип полноценного приложения. Особняком среди соревнований по программированию стоят соревнования по программированию искусственного интеллекта.

Соревнование по программированию искусственного интеллекта в большинстве случаев представляет собой разработку участниками стратегии в виде программного модуля для какой-либо игры, представленной в виде компьютерной программы [1, 2]. Разработанная стратегия должна успешно играть по описанным в игровом приложении правилам. Победители соревнования определяются в ходе поединков

стратегий участников друг с другом. Для зрелищности соревнования ход поединков транслируется.

Для проведения соревнований подходят следующие типы игр [2]:

- настольные игры (шашки, реверси и т.д.);
- динамичные игры (симуляция хоккейного матча, сборка каких-либо предметов в лабиринте и т. д.);
- стратегические игры с больших количеством объектов (имитация сражений армий и т. д.).

Задача разработки искусственного интеллекта является достаточно сложной и, во-многом, творческой. От участников требуется представить и предугадать ход матчей с соперниками, продумать множество различных вариантов развития событий, создать универсальную стратегию, которая будет боеспособна в любой ситуации, сложившейся на игровом поле. Однако еще более сложной задачей является разработка правил для таких соревнований и разработка системы для их проведения самими организаторами. Система для проведения соревнований, помимо возможности запуска двух стратегий и выдачи результата матча, должна выполнять множество других функций, таких, как визуальное отображение хода игры, помощь участникам в процессе отладке их стратегий, обработка большого количества исключительных ситуаций, которые могут возникнуть по ходу поединков. Из-за сложности разработки приложения для организации соревнований по программированию искусственного интеллекта такие соревнования проводятся достаточно редко, а их подготовка требует значительных ресурсов.

В данной работе рассматривается задача разработки фреймворка для организаторов соревнований по программированию искусственного интеллекта, который мог бы предоставить разработчику необходимый каркас для ускоренной реализации игровых приложений для соревнований, а также свести к минимуму вероятность совершения разработчиком ошибок, типичных для приложений в данной области.

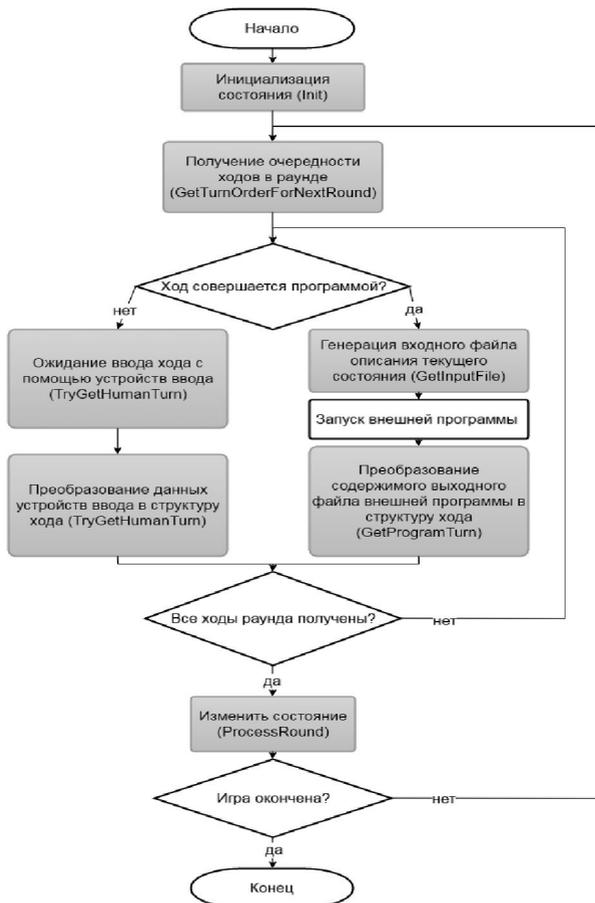
1. Реализация игры с использованием фреймворка

Разработанный фреймворк предполагает, что стратегии участников будут реализованы в виде отдельных программ. Будем называть такие программы *программами-стратегиями*. Программный модуль, который выполняется в рамках фреймворка и управляет ходом поединка между программами-стратегиями, будем называть *управляющей программой* (или *игровой программой*).

«Общение» между управляющей программой и программами-стратегиями происходит через текстовые файлы. Форматы таких файлов оговариваются в условиях соревнования. Каждый раз перед запуском программы-стратегии управляющая программа создает входной текстовый

файл с информацией о текущем состоянии игры. Программа-стратегия должна такой файл прочитать, «решить», какое действие/действия выполнять, и записать свое решение также в текстовый файл. Записанные команды, в случае их корректности, будут выполнены в игре управляющей программой [3]. Для облегчения отладки программ-стратегий должен быть предусмотрен «ручной» режим управления для одного из игроков.

На рис. 1 представлена схема работы основного цикла проведения поединка (матча) между программами-стратегиями.



Серым цветом выделены блоки, которые реализуются в управляющей программе

Рис. 1. Схема выполнения основного цикла поединка между стратегиями

Цикл проведения поединка заложен в ядре фреймворка. Из ядра вызываются методы, реализованные разработчиком в виде программного модуля (управляющей программы). Ниже приводится более подробное рассмотрение выделенных блоков и сигнатуры соответствующих методов:

- Инициализация состояния.

void Init(TStartSettings settings);

На данном этапе можно определить список участвующих игроков (программ-стратегий), их начальные очки, начальную позицию игровых объектов и так далее.

- Получение очередности ходов в раунде.

List<TPlayer> GetTurnOrderForNextRound();

Определяет и возвращает список игроков в том порядке, в котором они впоследствии должны быть вызваны ядром фреймворка для выполнения ходов.

– Блоки "Ожидание ввода хода с помощью устройства ввода", "Преобразование данных устройств ввода в структуру хода".

TTurn TryGetHumanTurn(TPlayer player, Input input);

Оба блока реализуются одним методом. Метод получает на вход экземпляр класса игрока и экземпляр класса для работы с устройствами ввода. Возвращает null в случае, если задание хода человеком еще не завершено, или заполненную структуру описания хода.

- Генерация входного файла описания текущего состояния.

string GetInputFile(TPlayer player);

Метод возвращает строку – содержимое текстового файла, создаваемого в директории программы-стратегии перед ее выполнением.

– Преобразование содержимого выходного файла внешней программы в структуру хода.

TTurn GetProgramTurn(TPlayer player, string output, ExecuteResult executionResult, string exitCode);

Принимает на вход данные о результате работы вызванной перед этим программы. Должен вернуть заполненную структуру описания хода.

- Изменить состояние.

void ProcessRound(TRound round);

Здесь совершается обработка совершенных игроками действий в соответствии с правилами игры.

Важно подчеркнуть, что приведенные выше методы должны быть реализованы программистом так, чтобы они всегда обрабатывали одинаково для конкретного набора входных данных.

Кроме приведенных на блок-схеме элементов в управляющем модуле должны быть реализованы два метода, отвечающих за визуализацию игрового состояния:

- Метод получения краткого описания текущего состояния игры в виде строки (например, текущий счет или набор оставшихся фигур у каждого из игроков). Краткое состояние игры отображается в окне выбора хранимого на сервере матча для просмотра.
- Метод отрисовки текущего игрового состояния.

2. Функциональность фреймворка

Фреймворк позволяет реализовывать системы проведения как очных, так и заочных соревнований. Во втором случае участники загружают свои стратегии на сервер, а администратор управляет запуском матчей и проведением турниров на сервере. Все данные об участниках и матчах сохраняются в базе данных сервера.

Локальная часть фреймворка, помимо организации основного цикла работы приложения, берет на себя реализацию ряда функций, которые схожи в различных управляющих программах для соревнований по программированию искусственного интеллекта. Эти функции предоставляются с возможностью их настройки разработчиком. Ниже приведены основные функции, реализованные во фреймворке:

- Основной цикл работы приложения с предоставлением разработчику готового каркаса приложения.
- Упрощена реализация графической составляющей визуализации матча. Для быстрой отрисовки объектов используется библиотека OpenGL, а для упрощения управлением графикой над библиотекой реализован паттерн Facade, предоставляющий простые и интуитивно понятные методы для отображения несложной графики.
- Полный функционал, связанный с просмотром матчей, такой, как возможность просмотра матча на разных скоростях, запись повтора, возможность воспроизведения с любого места.
- Пользовательский интерфейс со всеми необходимыми элементами управления (представлен на рис. 2, 3 и 4).
- Регистрация и авторизация участников.
- Возможность загрузки стратегий на сервер.
- Возможность проведения матчей и турниров на сервере.
- Возможность просмотра участниками соревнований матчей, проводимых на сервере, как после их окончания, так и в режиме реального времени (трансляция матчей с сервера).
- Сохранение информации обо всех прошедших матчах.
- Система контроля версий клиентского приложения и автозагрузка обновлений.

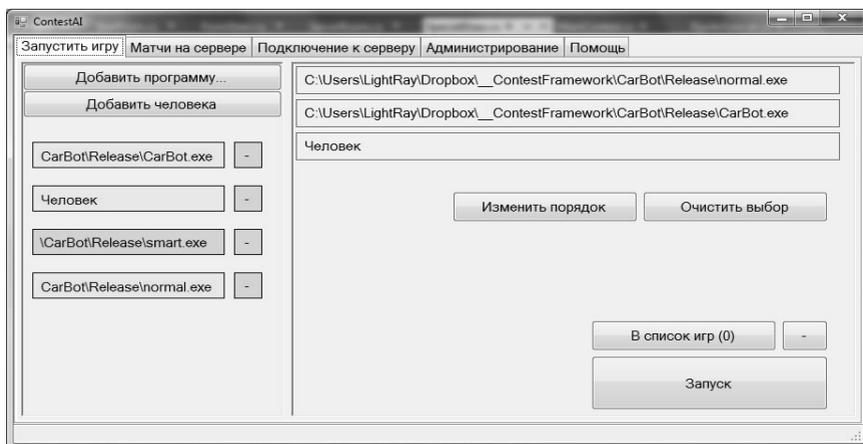


Рис. 2. Пользовательский интерфейс – вкладка «Запустить игру».

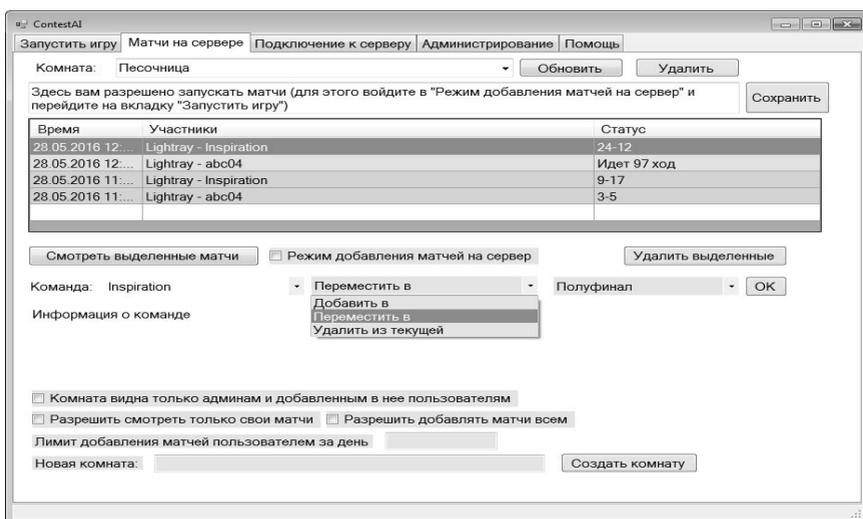


Рис. 3. Пользовательский интерфейс – вкладка «Подключение к серверу»

Каждая игра, реализованная с помощью рассматриваемого фреймворка, имеет свое визуальное представление. На рис. 4 представлено такое отображение для модификация известной игры на клеточной доске «Уголки».

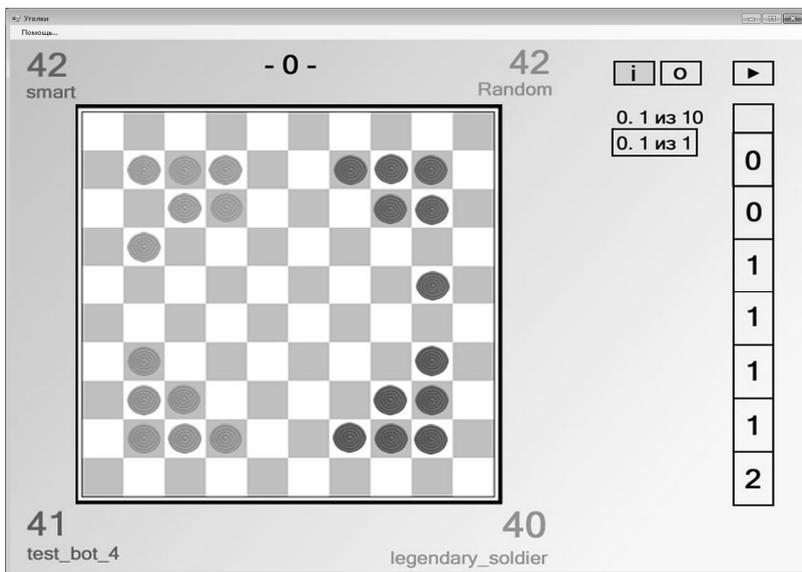


Рис. 4. Пользовательский интерфейс – игра «Уголки»

Заключение

Разработанный фреймворк позволяет значительно упростить как создание «игр» для соревнований по программированию искусственного интеллекта (например, реализация модификации игры «Уголки» от момента определения идеи до полной готовности к проведению соревнований заняла 6 часов), так и поведение таких соревнований (пакетное проведение поединков между стратегиями участников, заочные соревнования и т. д.). Данный фреймворк планируется использовать при проведении командных соревнований в рамках Марафона информационных технологий факультета компьютерных наук ВГУ.

Список литературы

1. DeNero J. Teaching introductory artificial intelligence with pac-man / J. DeNero, D. Klein // Proceedings of the Symposium on Educational Advances in Artificial Intelligence. – Pearson, 2010. – 91 p.
2. Крыжановский А. Code Game Challenge / А. Крыжановский. [Электронный ресурс] : сайт сообщества. – Режим доступа : <https://habrahabr.ru/post/71355/>
3. Титов А. Как устроен Code Game Challenge / А. Титов. [Электронный ресурс] : сайт сообщества. – Режим доступа : <https://habrahabr.ru/post/270953/>

ИНФОРМАЦИОННАЯ СИСТЕМА ЖИЗНЕОБЕСПЕЧЕНИЯ БИОЛОГИЧЕСКОЙ ТКАНИ

И.В. Капранчиков

Студент

Я.А. Туровский

Доцент

С.Д. Кургалин

Профессор

Введение

В настоящее время во всем мире активно развивается такая область междисциплинарной науки, как тканевая инженерия, основанная на процессе культивирования биологических тканей, в том числе трансплантируемых. Она является базой для более подробного изучения нервных тканей, создания биологических нейронных сетей, разработки различных нейроинтерфейсов. С ее помощью осуществляется исследование гибридных информационных систем с биологическими нейросетями и выполняются попытки их реализации.

В исследованиях такого рода важнейшим инструментом являются аппаратно-программные комплексы многоканального мониторинга клеточной активности, например, такие как MED64, USB-MEA256-systems и другие [1]. Отечественных аналогов таких комплексов пока еще не существует, и российские специалисты вынуждены закупать их за рубежом [2] или же использовать макроэлектроды, что крайне негативно сказывается на результатах проводимых исследований.

При этом комплексы иностранного производства обладают и рядом существенных недостатков:

- Высокой стоимостью, из-за чего большая часть отечественных научных коллективов не имеет возможности приобрести такое оборудование; к примеру, самый простой комплекс производства MEA-Systems стоит 10-15 млн. руб.;
- «Закрытостью» программного обеспечения и невозможностью применения в нем собственных программных решений;
- Невозможностью проверки соответствия указанных и реальных параметров устройств и комплектующих, что может привести к возникновению неконтролируемых ошибок в результатах исследований;
- Во многих иностранных комплексах (например, производства Alpha MED Scientific) используются мультиэлектродные матрицы с

изолирующим слоем на основе полиакриламида, а все вещества на его основе деградируют в водной среде, что с учетом толщины покрытия в 1,4 мкм ставит под вопрос надежность и долговечность матриц и достоверность получаемого сигнала.

1. Цель и задачи

Целью данной работы является разработка информационной системы жизнеобеспечения изолированной биологической (нервной, эпителиальной, соединительной или мышечной) ткани, что в дальнейшем позволит создать на ее основе полноценный многофункциональный программно-аппаратный комплекс для изучения и культивации биологической ткани. Для осуществления цели разработки было поставлено несколько задач:

- проектирование и создание готовых к использованию мультieleктродных матриц с различным количеством контактов и разной их конфигурацией;
- создание подсистемы терморегуляции;
- разработка подсистемы перфузии растворами;
- создание необходимого задела для последующих разработок других частей информационной системы.

2. Состав информационной системы и принципы ее использования

При проектировании системы были учтены особенности работы с живым образцом изолированной биологической ткани: значения жизненно-важных параметров должны быть максимально приближены к нормальным — значениям параметров, которыми обладает место происхождения образца ткани в целом организме; при значительных отклонениях от нормы образец погибает. Так же образец должен быть защищен от чужеродных для него микроорганизмов и вирусов, токсичных веществ, механических повреждений и вибраций, слишком сильного электрического тока (например, при электростимуляции образца ткани) [3, 4]. Таким образом, возникают требования к среде, в которую помещается изучаемый образец ткани [5]:

- защищенность от воздействия внешней среды и чужеродных образцу микроорганизмов;
- стабильный температурный режим среды около 37 градусов по Цельсию (для тканей человека);
- перфузия образца ткани растворами, обеспечение его кислородом и питательными веществами, своевременная замена или очистка раствора;
- минимально инвазивная регистрация данных с образца.

Информационная система состоит из следующих основных частей, это:

- мультиэлектродная матрица, контактирующая с нервной тканью;
- устройство регистрации сигналов, поступающих с нейрорезистивной матрицы;
- подсистема мониторинга и регуляции температурного режима образца ткани;
- подсистема перфузии образца ткани растворами.

Схема информационной системы представлена на рис. 1.

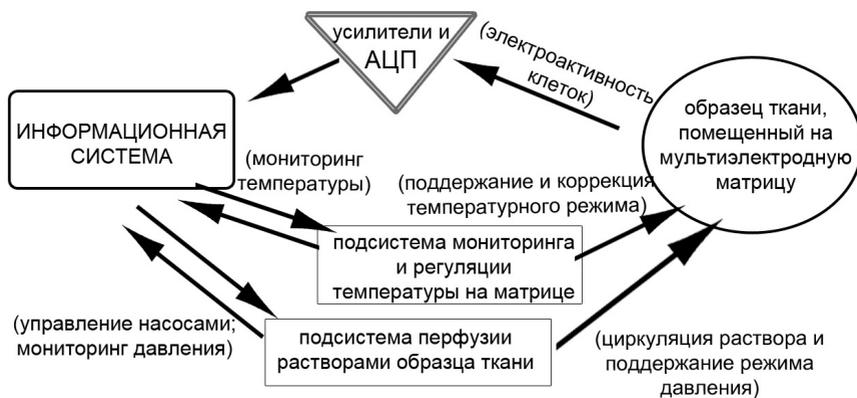


Рис. 1. Схема информационной системы

Программная часть информационной системы написана на языке С#.

Система функционирует следующим образом. Образец исследуемой ткани помещается на мультиэлектродную матрицу, сигнал с которой считывается многоканальным АЦП через усилители. Подсистема терморегуляции обеспечивает стабильность температуры. Подсистема перфузии обеспечивает снабжение образца ткани кислородом и питательными веществами путем поддержания циркуляции в системе специальных растворов.

Одной из важнейших частей системы является мультиэлектродная матрица. В отличие от других компонентов системы ее невозможно приобрести в нашей стране. Нами планируется применять в одном комплексе матрицы с разным числом контактов и их конфигураций — 16, 32, 64 и т. д. до 256 включительно. Прототипы матриц изготавливаются несколькими методами: а) путем фотолитографии из диэлектрической пластины с серебряным покрытием; б) с помощью фоторезиста на основе меди с последующим серебрением; в) из серебряных нитей путем электрохимической «заточки» изготавливаются одиночные электроды и из них уже вручную под микроскопом формируется матрица.

Разрешение (толщина дорожек) прототипов матриц – 100 мкм, но с помощью более совершенных материалов можно изготовить матрицу с разрешением до 10 мкм. Одиночные электроды удается стабильно «заточивать» до 25 мкм. Фотография центрального фрагмента прототипа матрицы на 32 контакта представлена на рис.2.

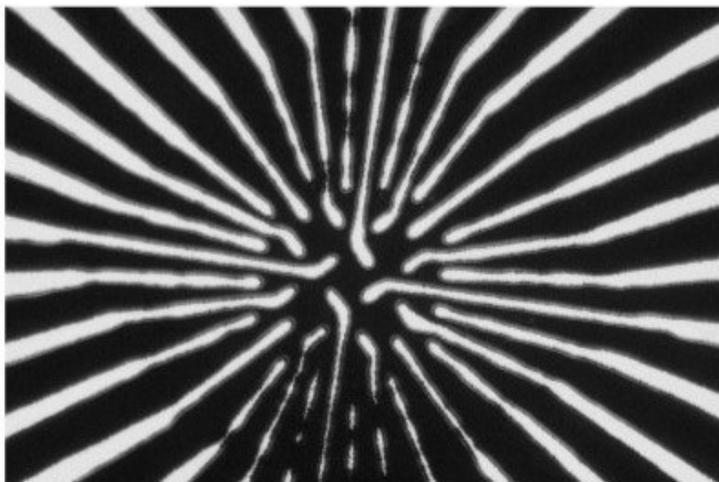


Рис. 2. Центральный фрагмент прототипа мультиэлектродной матрицы под микроскопом

Сигнал с электродов в данной системе считывается с помощью многоканального АЦП.

Подсистема терморегуляции состоит из изолирующей капсулы, внутри которой с помощью термисторов поддерживается необходимая температура. Информационная система обрабатывает сигнал с термодатчиков, определенным образом расположенных во внешней термокапсуле, и регулирует напряжение на термисторах, увеличивая или уменьшая выделение тепла. Для равномерности распределения потоков нагретого газа в капсуле используется, перемешивая их, вентилятор, работа которого так же контролируется подсистемой.

Подсистема перфузии включает в себя герметичный контур, находящийся внутри термокапсулы. Она состоит из микроэлектродной капсулы, в которой находится образец ткани, контактирующий с микроэлектродами, резервуара с раствором, насоса, обеспечивающего равномерную циркуляцию раствора и подключенного шлангами к капсуле, датчика давления и дополнительного насоса, служащего для регулировки давления.

Заключение

В настоящей работе создана информационная система, которая обеспечивает:

- снабжение образца исследуемой биологической ткани питательными веществами и кислородом;
- поддержание стабильного температурного режима и, при необходимости, его оперативную коррекцию;
- поддержание требуемого давления, своевременную коррекцию изменения давления;
- считывание и первичную обработку сигнала электрической активности клеток биологической ткани стандартными методами.

Разработанная система является основой для проведения дальнейших разработок многофункционального программно-аппаратного комплекса для изучения и культивации биологической ткани.

Список литературы

1. Thiebaud P. Microelectrode arrays for electrophysiological monitoring of hippocampal organotypic slice cultures / P. Thiebaud, M. Koudelka-Hep, L. Stoppini // IEEE Trans. Biomed. Eng. – 1997. – V. 44. – P. 1159-1163.
2. Мухина И.В. Мультиэлектродные матрицы – новые возможности в исследовании пластичности нейрональной сети / И.В. Мухина, В.Б. Казанцев, Л.Г. Хаспеков [и др.] // Современные технологии в медицине. – 2009. – № 1. – С. 8-15.
3. Li Y. A Multichannel Waveform Generator for Spatiotemporal Stimulation of Dissociated Neuronal Network on MEA / Y. Li, H. Li, Y. Wang // Journal of Medical and Bioengineering. – 2014. – P. 4-5.
4. Wagenaar D.A. A versatile all-channel stimulator for electrode arrays, with real-time control / D.A. Wagenaar, S.M. Potter // Journal of Neural Eng. – 2004. – № 1. – P. 39-45.
5. Potter S.M. A new approach to neural cell culture for long-term studies / S.M. Potter, T.B. DeMarse // Journal of Neurosc. Methods. – 2001. – № 110. – P. 17-24.

АЛГОРИТМ НОРМАЛИЗАЦИИ УРАВНЕНИЙ МНОГОМЕРНЫХ ПРОСТРАНСТВ

Д.А. Капустина

Студент

А.В. Атанов

Ассистент

Введение

Идея преобразования объектов к удобному в каком-либо смысле виду перед их непосредственным изучением является одной из центральных в математике. Простейшим примером здесь может служить, например, приведение дифференциальных уравнений в частных производных к каноническому виду.

Похожие идеи активно используются и в комплексной геометрии — в частности, в задаче поиска и классификации однородных поверхностей многомерных пространств. Существует три основных подхода к решению этой задачи: дифференциально-геометрический, алгебраический и коэффициентный. В данной работе описывается реализация алгоритма нормализации по Мозеру уравнений многомерных комплексных пространств, основывающегося на коэффициентном подходе.

1. Постановка задачи

Основным объектом изучения в работе являются уравнения вещественных аналитических гиперповерхностей трёхмерного комплексного пространства (в частности, возникающие в ряде нерешённых до сих пор задач комплексной геометрии, связанных с изучением однородных гиперповерхностей). В ряде основополагающих работ по данной тематике показано, что такие поверхности удобнее всего изучать, представляя их в нормальной форме Мозера (см. [1]):

$$v = z_1 z_1 + z_2 z_2 + \sum_{k,l \geq 2, m \geq 0} N_{klm}(z, z) u^m, \quad (1)$$

где N_{klm} — однородные многочлены степени k по переменным $z = (z_1, z_2)$, l — по переменным $\bar{z} = (\bar{z}_1, \bar{z}_2)$; $u = \operatorname{Re}(w)$, $v = \operatorname{Im}(w)$.

В данной работе изучаются однородные вещественные гиперповерхности. Как показано в [2], любая такая гиперповерхность полностью определяется следующими слагаемыми её уравнения Мозера: $N_{220}(z, \bar{z})$, $N_{320}(z, \bar{z})$, $N_{420}(z, \bar{z})$, $N_{330}(z, \bar{z})$, $N_{221}(z, \bar{z})$, $N_{222}(z, \bar{z})$, $N_{321}(z, \bar{z})$.

При этом каждый полином $N_{klm}(z, \bar{z})$ является элементом специального подпространства \mathfrak{R}_{kl} . Например, подпространство \mathfrak{R}_{22} пятимерно, а его базис составляют многочлены:

$$E_0 = |z_1|^4 - 4|z_1|^2|z_2|^2 + |z_2|^4, \quad E_1 = z_1^2 z_2^2 + z_2^2 z_1^2, \quad E_2 = i(z_1^2 z_1^2 - z_2^2 z_1^2), \\ E_3 = (z_1 z_2 + z_2 z_1)(|z_1|^2 - |z_2|^2), \quad E_4 = i(z_1 z_2 + z_2 z_1)(|z_1|^2 - |z_2|^2).$$

Уравнение произвольной поверхности, естественно, изначально не имеет нужного вида (то есть оно не представлено в форме Мозера). Для того чтобы привести его к форме Мозера, требуется выполнить процедуру нормализации, что представляет собой достаточно сложную техническую задачу. В некоторых простейших случаях уравнение поверхности можно привести к форме Мозера «вручную», однако в общем случае этот процесс оказывается настолько сложным, что возникает естественная необходимость в компьютерных вычислениях.

Основной задачей данной работы является разработка и реализация компьютерного алгоритма приведения уравнения однородной поверхности из заданного класса к нормальной форме Мозера.

2. Алгоритм нормализации по Мозеру

Уравнение жёсткой вещественной поверхности M трёхмерного комплексного пространства имеет вид:

$$v = z_1 z_1 + z_2 z_2 + \sum_{k, l \geq 2, k+l \geq 3} F_{kl}(z, \bar{z}). \quad (2)$$

Здесь $F_{kl}(z, \bar{z})$ — однородный полином степени k по переменным $z = (z_1, z_2)$ и степени l по переменным $\bar{z} = (\bar{z}_1, \bar{z}_2)$, $v = \text{Im}(w)$.

Разработанный компьютерный алгоритм является реализацией известного алгоритма нормализации, теоретическая часть которого описана в [2]. Рассмотрим его основные шаги.

Шаг 1. Замена переменных:

$$w^* = w - 2i \sum_{k > 2} F_{k0}(z, \bar{z}) \quad (3)$$

обнуляет слагаемые F_{k0} и F_{0k} , сохраняя при этом все остальные слагаемые уравнения (2).

Шаг 2. Слагаемые F_{k1} и F_{1k} в (2) удаляются (см. [1]) с помощью замены:

$$z_j^* = z_j + Q_j + T_j + C_j + \dots, \quad j = 1, 2, \quad (4)$$

где $Q_j = q_{j1} z_1^2 + q_{j2} z_1 z_2 + q_{j3} z_2^2$; $T_j = t_{j1} z_1^3 + t_{j2} z_1^2 z_2 + t_{j3} z_1 z_2^2 + t_{j4} z_2^3$ и т.д.

Однако уравнение (2) записано в переменных z , а не z^* , поэтому последнюю замену (4) напрямую использовать нельзя. Требуется найти обратную к ней замену, то есть фактически выразить z^* через z :

$$z_j = z_j^* + \varphi_{j2} + \varphi_{j3} + \varphi_{j4} + \dots, j = 1, 2, \quad (5)$$

где φ_{jk} — однородные полиномы, зависящие от переменных z_1^*, z_2^* .

Записав очевидное тождество:

$$\left\{ -z_j + z_j^* + \varphi_{j2} + \varphi_{j3} + \varphi_{j4} + \dots \right\} \Big|_{x_j^* = x_j + Q_j + T_j + C_j + \dots} \equiv 0 \quad (6)$$

где $j = 1, 2$, можно выделить в его левой части слагаемые степеней 1, 2, 3, ..., и, приравнявая их нулю, однозначно определить все полиномы φ_{jk} .

Отметим, что на этом этапе возникают самые громоздкие вычисления. После замены (4) переменные z_j^* становятся полиномами и при подстановке их в полиномы φ_{j2} совокупные степени получающихся многочленов оказываются настолько большими, что, например, система Maple, используемая для компьютерной реализации данного алгоритма, не справляется с такой подстановкой уже на самых младших слагаемых.

Однако, как было указано ранее, для изучения однородных поверхностей достаточно рассматривать компоненты, имеющие степень не выше шестой. Поэтому при реализации данного алгоритма производилась автоматическая «редукция» степеней до 6-й включительно в каждом из слагаемых тождества (6). Это позволило существенно ускорить вычисления и однозначно определить полиномы φ_{jk} , а значит и всю замену, удаляющую из уравнения (2) многочлены F_{1k} .

После выполнения шага 2 алгоритма уравнение (2) преобразуется к виду:

$$v = z_1 z_1 + z_2 z_2 + H_{22} + (H_{32} + H_{23}) + (H_{42} + H_{33} + H_{24}) + \dots \quad (7)$$

Здесь $H_{kl}(z, z)$ — однородный полином степени k по переменным $z = (z_1, z_2)$ и степени l по переменным $z = (z_1, z_2)$, $v = \text{Im}(w)$.

Шаг 3. Как видно, уравнение (7) уже достаточно близко к нормальной форме Мозера (1), однако, для окончательного перехода (точнее, проектирования полиномов H_{kl} в пространства \mathfrak{R}_{kl}) необходимо выполнить ещё одно «нормализующее» преобразование координат, имеющее вид:

$$\begin{cases} z_j^* = z_j + f_{(3)}^j + f_{(4)}^j + f_{(5)}^j + \dots, j = 1, 2 \\ w^* = w + g_{(4)} + g_{(5)} + \dots \end{cases} \quad (8)$$

Здесь $f_{(k)}^j, g_{(l)}$ — однородные полиномы, зависящие от переменных z_1, z_2, w .

Очевидно, что при осуществлении замены (8) все проблемы, связанные с существенным возрастанием степеней получающихся полиномов, описанные на шаге 2, сохраняются и даже усугубляются. Однако для рассматриваемых жёстких уравнений в [2] выписаны конкретные формулы перехода от полиномов $H_{kl}(z, z)$ к нормализованным полиномам из уравнения (1). Используя их, получаем в итоге искомый вид нормального уравнения Мозера:

$$v = z_1 z_1 + z_2 z_2 + N_{220} + (N_{320} + N_{230}) + (N_{420} + N_{330} + N_{240}) + \dots \quad (9)$$

3. Примеры нормализованных по Мозеру уравнений

Классификацию аффинно-однородных поверхностей трёхмерного вещественного пространства, приведённую в [3], можно использовать для описания некоторого семейства голоморфно-однородных поверхностей трёхмерного комплексного пространства. Для проверки работы нашего алгоритма были рассмотрены два семейства таких поверхностей.

Пример 1. Определяющие компоненты нормального уравнения Мозера для семейства однородных поверхностей:

$$ix_1 = x_2^2 - x_1^\alpha, 1 < \alpha < 2$$

имеют вид:

$$\begin{aligned} N_{22} &= \frac{\alpha + 2}{24(\alpha - 1)(\alpha - 2)} E_0 - \frac{1}{4(\alpha - 2)} E_1. \\ N_{32} &= -\frac{(\alpha - 1)^4 (\alpha - 2)^4}{24\mu^{9/2}} z_1^3 z_2^2 - \frac{(\alpha - 1)^3 (\alpha - 2)^2 (2\alpha - 3)}{12\mu^{7/2}} z_1^3 z_2^2 + \\ &+ \frac{(\alpha - 1)^4 (\alpha - 2)^3 (\alpha + 2)}{12\mu^{9/2}} z_1^2 z_2 z_1 z_2 - \frac{(\alpha - 1)^3 (\alpha - 2)^2}{4\mu^{7/2}} z_1 z_2^2 z_1^2 + \\ &+ \frac{(\alpha - 1)^4 (\alpha - 2)^3 (\alpha - 10)}{24\mu^{9/2}} z_1 z_2^2 z_2^2 + \frac{1}{2\mu^{3/2}} z_2^3 z_1 z_2. \\ N_{42} &= -\frac{(2\alpha - 3)(\alpha - 3)}{48\nu} z_1^4 z_1^2 + \frac{(2\alpha - 3)(3\alpha - 5)}{48\nu} z_1^4 z_2^2 + \frac{2\alpha - 3}{6\nu} z_1^3 z_2 z_1 z_2 + \\ &+ \frac{1}{8(\alpha - 2)^2} z_1^2 z_2^2 z_1^2 - \frac{2\alpha - 3}{8\nu} z_1^2 z_2^2 z_2^2 - \frac{1}{2\nu} z_1 z_2^3 z_1 z_2 - \\ &- \frac{\alpha + 1}{16(\alpha - 1)^2 (\alpha - 2)} z_2^4 z_1^2 + \frac{1}{16\nu} z_2^4 z_2^2. \\ N_{33} &= -\frac{(\alpha - 3)(\alpha^4 - 4\alpha + 9)}{36\omega} z_1^3 z_1^3 + \frac{\alpha + 3}{12\nu} z_1^3 z_1 z_2^2 + \end{aligned}$$

$$\begin{aligned}
& + \frac{\alpha^4 - 2\alpha + 5}{4\omega} z_1^2 z_2^2 z_1^2 z_2 - \frac{1}{4\nu} z_1^2 z_2^2 z_2^3 + \frac{\alpha + 3}{12\nu} z_1 z_2^2 z_1^3 - \\
& - \frac{\alpha + 1}{4\omega} z_1 z_2^2 z_1 z_2^2 - \frac{1}{4\nu} z_2^3 z_1^2 z_2 + \frac{1}{4\omega} z_2^3 z_2^3.
\end{aligned}$$

Здесь $\mu = -(\alpha - 1)(\alpha - 2)$, $\nu = (\alpha - 1)(\alpha - 2)^2$, $\omega = (\alpha - 1)^2(\alpha - 2)^2$.

Пример 2. Определяющие компоненты нормального уравнения Мозера для семейства однородных поверхностей:

$$\nu = (y_2 + x_1^2 + \gamma y_1^2)^{3/2}, \gamma < -1$$

имеют вид:

$$\begin{aligned}
N_{22} &= \frac{1}{6} E_0, \quad N_{32} = -\frac{\sqrt{3}}{18} (3z_1^2 z_2^2 z_1^2 - 6z_1 z_2^2 z_1 z_2 + z_2^3 z_2^2), \\
N_{42} &= \frac{1}{18} (6z_1^2 z_2^2 z_1^2 - 8z_1 z_2^3 z_1 z_2 + z_2^4 z_2^2), \\
N_{33} &= -\frac{1}{27} (z_1^3 z_1^3 - 15z_1^2 z_2^2 z_1^2 z_2 + 21z_1 z_2^2 z_1 z_2^2 - 3z_2^3 z_2^3).
\end{aligned}$$

Заключение

Основным результатом данной работы стала компьютерная реализация алгоритма нормализации однородных поверхностей трёхмерного комплексного пространства. С использованием разработанного алгоритма была проведена нормализация по Мозеру для двух семейств аффинно-однородных гиперповерхностей.

Список литературы

1. Chern S.S. Real hypersurfaces in complex manifolds / S.S. Chern, J.K. Moser // Acta Mathematica. – 1974. – V. 133. – № 1. – P. 219-271.
2. Лобода А. В. Однородные строго псевдо-выпуклые гиперповерхности в \mathbb{C}^3 с двумерными группами изотропии / А. В. Лобода // Матем. сборник. – 2001. – Т. 192. – С. 3-24.
3. Doubrov B. M. Homogeneous surfaces in the three-dimensional affine geometry / B.M. Doubrov, B.P. Komrakov, M. Rabinovich // Geometric Topology of Submanifolds VIII, World Scientific. – 1996. – P. 168-178.

РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ МАГАЗИНОВ РОЗНИЧНОЙ СЕТИ ООО «ФОРМАТ»

В.В. Карнович

Студент

Н.А. Тюкачев

Доцент

Введение

В современном мире, в эпоху информационных технологий, в любой сфере деятельности люди ищут способы внедрения различных программных продуктов для увеличения производительности. Всё развивается на ускорение. Желания у людей увеличиваются, следовательно, растут и требования к обеспечению рабочего места инструментами оперативного решения задач. Для быстрой и качественной работы необходимо совершенствовать программное обеспечение, переходить с аналоговых методов на цифровые технологии, автоматизировать процесс работы, как можно больше.

1. Цель работы

Целью данной работы является создание программного продукта для розничной сети центрально-черноземного региона ООО «Формат», которое оптимизирует и автоматизирует работу сотрудников магазинов компании «на местах», позволит оперативнее решать текущие задачи по формированию ежедневной отчетности. На данный момент имеется потребность в создании единой базы данных, включающей в себя сведения о работающих сотрудниках магазинов, графиках работы, схемах расчета зарплаты и т. д. Работа с данными будет осуществляться с помощью специального приложения, которое позволит найти контактные данные других магазинов и сотрудников сети (например, для оперативного информирования о рекламных акциях, смене графика работы в праздничные дни, изменении цен в преysкyранте или коэффициентов при расчете заработной платы и т. д.). Также можно будет проверить или изменить свой график работы (ежегодный отпуск, учеба, отпуск без содержания), заполнить и сохранить отчет для ревизионной комиссии, оперативно вычислить заработную плату за любой промежуток времени.

2. Постановка задачи

Для достижений поставленной ранее цели необходимо решить следующие задачи:

- провести анализ существующих программных средств, используемых в ООО «Формат»;
- выявить способы оптимизации рабочего процесса;
- создать базу данных;
- создать непосредственно приложение, которое позволит решать несколько специализированных задач:
 - составление ежемесячного табеля работы сотрудников, сохранение в виде электронной таблицы Excel и отправка по электронной почте в ревизионный отдел;
 - составление ежедневного отчета по выручке (также с возможностью сохранения в Excel и отправки электронной почтой в ревизионный отдел);
 - расчет заработной платы за каждый день отдельно по данным из ежедневных отчетов по выручке;
 - просмотр и редактирование контактной информации о сотрудниках магазина.

Требования к разрабатываемой программе:

- программный интерфейс должен быть понятен любому пользователю;
- клиентская часть должна быть связана с базой данных, способна обновлять и получать всю необходимую информацию из базы данных;
- база данных должна содержать в себе актуальную информацию;
- возможность редактирования информации в базе данных.

Конечными пользователями приложения являются сотрудники магазинов компании.

3. Анализ задачи

В настоящее время в магазинах ООО «Формат» существует ряд задач, которые нуждаются в доработке. Существует большая необходимость в объединении различных отчетов в единую систему во избежание многократного заполнения одних и тех же данных и произведения аналогичных расчетов.

Таким образом, необходимо создать удобное приложение, которое смогло бы решать несколько важных задач оперативно и комплексно.

На рынке существует множество программ и сайтов, которые могут решать задачи, рассмотренные в п. 2 настоящей статьи. Некоторые из них представлены в таблице, в которой проведен сравнительный анализ необходимых требований, выдвигаемых к разрабатываемой программе, и возможности уже существующих аналогов. Таким образом, если ячейка таблицы отмечена плюсом, значит, программа удовлетворяет заданным требованиям.

Сравнение существующих аналогов

Название программы	Требования, выдвигаемые к разрабатываемой программе					
	Информация о сотрудниках	Расчет заработной платы	Сохранение в Excel	Отправка по почте	График работы	Бесплатная версия
AFM: Scheduler 1/11	+	-	-	-	+	+
WinForms Scheduler	+	-	-	-	+	-
CakeHR	+	-	+	+	+	-
1С:Зарплата и Управление персоналом	+	+	-	-	-	-
Контур.Бухгалтерия	-	+	+	-	-	-
Зарплата и кадры Инфо-предприятие	+	+	+	+	-	-
Контур.Зарплата	-	+	-	+	+	-
Бухсофт: Зарплата и кадры 2016	+	+	-	+	+	-
Разрабатываемая программа	+	+	+	+	+	+

Из таблицы видно, что большинство существующих продуктов являются специализированными, то есть не универсальными в решении поставленных требований, но, как правило, дорогостоящими, и при этом сложными в освоении и имеющими избыточные функции для решения задач на уровне магазина. Поэтому в компании приветствуется разработка своего «бюджетного» программного продукта для решения задач сотрудников. Так, например, для работы с кассой и базой данных по товару, уже используется специально созданная программа под названием «Фотоучёт».

4. Программная реализация

Требования к составу и параметрам технических средств обусловлены тем, что база данных находится на локальном сервере компании, а

приложение находятся на компьютерах компании в магазинах. Поэтому требования средств определен следующим образом:

- сервер базы данных MySQL;
- среда разработки Visual Studio 2015;
- язык программирования C#.

Для реализации поставленных задач программа должна иметь следующие возможности:

- хранение данных, которые используются при формировании отчетов;
- обмен данными между приложением и базой данных;
- изменение результативных данных, при изменении исходных параметров;
- отображение всех отчетов в виде таблиц.

Вся информация, представленная в таблицах разработанного приложения, хранится в реляционной свободной системе управления базой данных MySQL, которая является бесплатной, быстрой и простой в использовании СУБД, и она прекрасно подходит для разработки небольшого программного приложения, представленного в данной статье. Проектирование и реализация баз данных в СУБД MySQL проходило с помощью специально программного обеспечения – MySQL Workbench [1].

Часть разработанной схемы баз данных показана на рис. 1.



Рис. 1. Схема баз данных

Для реализации клиентского приложения написанного на языке программирования С# в программном коде использовалось несколько классов, которые можно разделить на несколько функциональных групп:

- классы для алгебраических действий;
- классы для подготовки данных перед сохранением в Excel;
- классы для описания свойств компоненты datagridview разрабатываемой программы;
- классы для работы с базой данных:
 - открытие и закрытие соединения с базой данных;
 - загрузка информации из базы данных;
 - удаление информации из базы данных;
 - очистка базы данных.

Далее приведен пример одной из функциональных групп. Классы для работы с базой данных представлены на рис. 2.

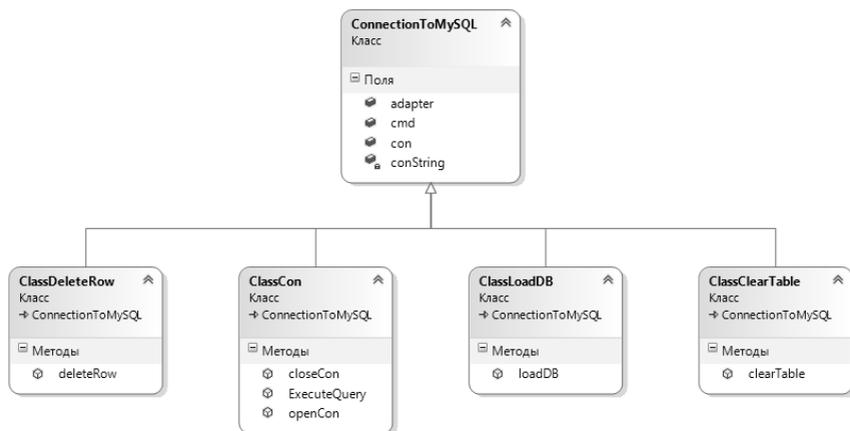


Рис. 2. Диаграмма классов разрабатываемого программного продукта отвечающие за работу с базой данных

Класс *ConnectionToMySQL* содержит все необходимые поля, которые нужны для соединения с базой данных. Данный класс имеет четыре класса потомка, содержащие методы для различных взаимодействий с базой данных. Рассмотрим каждый класс-потомок подробнее.

Класс *ClassDeleteRow* необходим для реализации процесса удаления выделенной строки. Содержит в себе единственный метод описывающий данное действие. Класс *ClassCon* решает задачу подключения и проверки соединения с базой данных. Класс *ClassLoadDB* обеспечивает выгрузку данных в приложение из определенной таблицы базы данных. Класс

ClassClearTable осуществляет возможность очистки всех строк таблицы баз данных [2].

5. Пользовательский интерфейс

Стартовое окно приложения включает несколько вкладок для взаимодействия с таблицами отчетов. Вкладки содержат редактируемые таблицы, например, при расчете заработной платы можно поменять значения коэффициентов, в результате чего произойдет перерасчет конечных значений. Решена задача создания отчетов, их сохранения в файл Excel и отправки на электронную почту ревизионного отдела. Так же пользователи программы смогут ознакомиться с актуальной контактной информацией о сотрудниках магазина, посмотреть и исправить график расписания за текущий месяц, рассчитать заработную плату за конкретный день месяца. Стартовое окно приложения изображено на рис. 3.

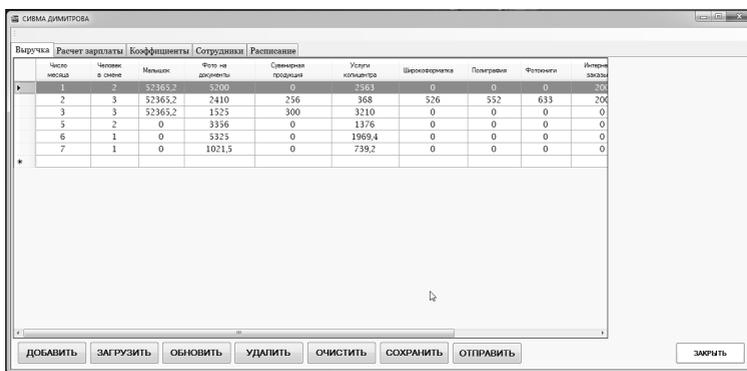


Рис. 3. Стартовое окно приложения

Пример сохраненного файла в Excel – на рис. 4.

	A	B	C	D	E	F	G
1	Дата заполнения:	27.05.2016					
2	Число месяца	1	2	3	5	7	10
3	Мальшок	52365,2	52365,2	52365,2	0	0	10
4	Фотоуслуги	15105,94	-28944,64	6028	15979,29	22697,42	25420
5	Фото на документы	5200	2410	1525	3356	1021,5	10
6	Сувениры	0	256	300	0	0	10
7	Услуги копицентра	2563	368	3210	1376	739,2	10
8	Широкоформатка	0	526	0	0	0	10
9	Полиграфия	0	552	0	0	0	10
10	Фотокниги	0	633	0	0	0	10
11	Интернет заказы	200	200	0	0	0	10
12	Фототовары	2563,36	42000	4200	0	0	10
13	ИТОГО	77997,5	70365,56	67628,2	20711,29	24458,12	25510

Рис. 4. Пример сохраненного отчета в файле Excel

Заключение

Данная статья посвящена проблеме недостаточной автоматизации рабочего процесса на примере магазинов розничной сети ООО «Формат». Описано создание программного приложения для решения «внутренних» текущих задач сотрудников. Вследствие внедрения описанного продукта в магазинах сети отпала необходимость использования нескольких программ при заполнении и отправки отчетов, были минимизированы ошибки при составлении отчетов, связанные с человеческим фактором. Использование нового продукта помогает оперативно решать ряд задач, связанных с отчетностью перед ревизионным отделом, вопросы по расчету заработной платы, а также появилась возможность для быстрого доступа ко всем необходимым контактными данным сотрудников магазина ООО «Формат».

Список литературы

1. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench : Учебное пособие / С.А. Мартишин [и др.]. – М. : ИД ФОРУМ : НИЦ Инфра-М, 2012. – 160 с.
2. Working with Class Diagrams (Class Designer) [Электронный ресурс]. – Режим доступа : <https://msdn.microsoft.com/ru-ru/library/33864ckt.aspx>

ПОИСК ПОХОЖИХ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ НА ОСНОВЕ АНАЛИЗА АУДИОСИГНАЛА

А. С. Колесов

Студент

Д. И. Соломатин

Старший преподаватель

Введение

Музыка окружает нас повсюду. Мы слушаем ее на работе и на отдыхе. Даже на улице и в общественном транспорте до нас доносятся ее звуки. Она является неотъемлемой частью нашей жизни.

Разным людям нравится разная музыка. Как и в любом другом виде искусства, у каждого человека существует свой индивидуальный набор интересов и предпочтений. Это могут быть любимые группы, альбомы или композиции. Именно поэтому люди изучают сотни композиций в поисках тех, которые войдут в список их любимых произведений.

Однако здесь людям способны помочь рекомендательные системы. Их задача заключается в том, чтобы, имея информацию об интересах человека, предложить ему для дальнейшего изучения только то, что может заинтересовать его с наибольшей вероятностью.

1. Подходы к разработке рекомендательных систем

Существует два основных подхода к разработке рекомендательных систем: коллаборативная и контентная фильтрация [1].

Коллаборативная фильтрация вырабатывает рекомендации на основе поиска тех пользователей, которые демонстрируют схожие предпочтения или шаблоны поведения. К примеру, если первый пользователь читал статьи про Debian и Ubuntu, а второй интересовался Ubuntu, система определит, что интересы этих пользователей схожи и посоветует второму пользователю ознакомиться с материалами по Debian. Основным допущением данной модели является предположение о том, что пользователи, которые оценивали что-либо одинаково в прошлом, склонны давать схожие оценки и в будущем.

Контентная фильтрация формирует рекомендации на основе анализа поведения пользователя в прошлом. Если какой-либо пользователь часто слушал The Beatles, то контентная фильтрация может использовать эту информацию для поиска подобного контента и предложения такого контента в качестве рекомендованного для этого пользователя (записи The Beatles, которые он еще не слушал, или музыка похожих групп). Этот контент может быть определен вручную или подобран автоматически на основе различных методов подобия.

2. Подходы к определению подобия аудиозаписей

Обычно алгоритмы поиска похожей музыки изучают и сравнивают не сам аудиосигнал, а метаинформацию, содержащую в себе краткие сведения о произведении, такие как жанр и информацию об исполнителе, группе или композиторе.

Возможно, данный подход используется из-за его относительной простоты реализации, но, если поразмышлять, секунда аудиосигнала содержит в себе в тысячи раз больше информации, чем страница печатного текста. Соответственно, при его изучении, можно узнать гораздо больше того, что содержится в его описании. Например, информацию о темпе, ритме, форме произведения, а так же о составе инструментов, который его исполнял.

К тому же, поиск похожей музыки на основе анализа сигнала способен давать более точные рекомендации из-за того, что он, в отличие от стандартного подхода, более непредвзято относится к жанровому делению. К примеру, некоторые современные композиции демонстрируют

значительное сходство с классическими произведениями, однако не всякая рекомендательная система имеет об этом представление.

На первый взгляд, преимущества использования вышеописанного подхода очевидны. В данной работе проводится исследование по созданию алгоритма поиска похожей музыки на основе анализа аудиосигнала, а так же оценке его эффективности при составлении рекомендаций.

3. Алгоритм анализа сигнала

Задача поиска похожих объектов, по сути, является задачей сортировки объектов, среди которых осуществляется поиск, по величине, определяющей меру схожести с объектом, для которого осуществляется поиск. Для того чтобы существовала возможность сортировки множества, необходимо, чтобы на этом множестве была определена операция сравнения. Из этого следует, что достаточно разработать алгоритм, способный определять меру схожести двух поданных на вход записей.

Для определения подобия можно использовать такие характеристики, как темп, ритм, жанр, форму произведения, состав инструментов и их тембр. Так как алгоритм является прототипом, а некоторые из перечисленных параметров, такие как жанр и инструментальный состав, сложны для определения, для экспериментов ограничимся определением темпа и нескольких параметров, характеризующих средний спектр произведения.

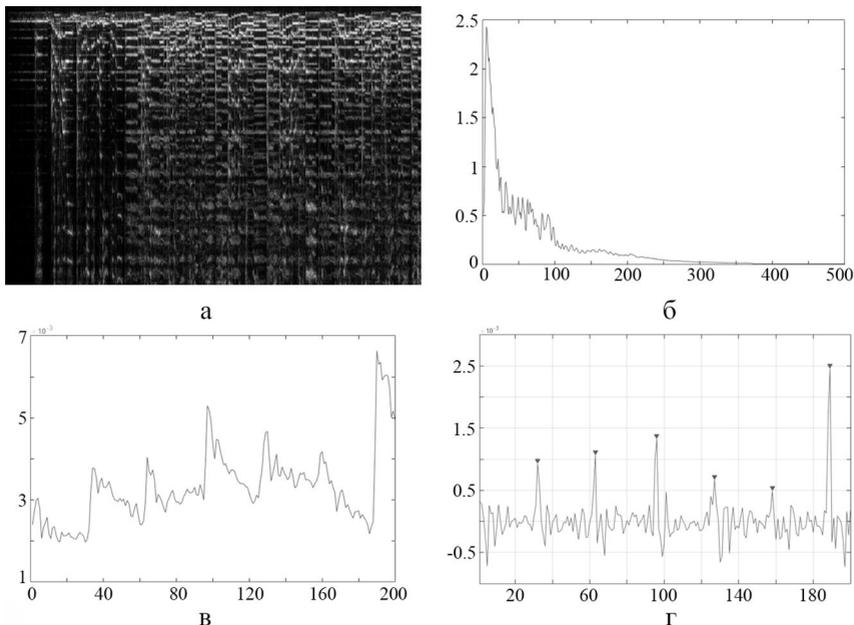
Рассмотрим одну из записей на входе. Аудиосигнал сложен для анализа в том виде, в котором он есть, поэтому применим к нему быстрое преобразование Фурье. Получившееся распределение интенсивности от времени и частоты носит название спектрограммы (см. рис. 1а).

Для определения темпа произведения просуммируем спектрограмму по оси частот и получим зависимость спектральной мощности сигнала от времени, называемой потоком спектра (см. рис. 1в).

На графике можем наблюдать точки, в которых функция резко возрастает. Они соответствуют моментам времени, когда музыкант бьет по струнам гитары. В свою очередь эти моменты соответствуют сильным долям тактов произведения. Так как нас интересуют точки роста, используем производную данной функции (см. рис. 1 г), чтобы выделить их. Теперь, зная местоположение во времени сильных долей, посчитаем их количество и разделим на продолжительность всего произведения. Полученная величина характеризует темп произведения [3].

Теперь найдем среднее значение спектрограммы по времени. Полученное распределение является средним спектром сигнала (см. рис. 1 б). Для экспериментов используем такие его характеристики как среднее значение на всем диапазоне частот, а так же средние значения на

интервалах для низких (20 Гц – 4 кГц), средних (4 кГц – 7 кГц) и высоких (7 кГц – 10 кГц) частот.



а – спектрограмма, б – средний спектр, в – поток спектра, г – производная потока

Рис. 1. Промежуточные результаты анализа аудиосигнала

Таким образом, результатом вышеописанных вычислений имеем шесть величин, характеризующих сигнал: длительность, темп, средние значения спектра в диапазоне низких, средних, высоких частот и на всем диапазоне.

Теперь, вычислив характеристики для двух сигналов, определим меру схожести сигналов следующим образом:

$$D_{12} = 2\Delta T + \Delta S_a + \frac{1}{2} \Delta S_l + \frac{1}{2} \Delta S_m + \frac{1}{2} \Delta S_h + \frac{1}{3} \Delta L,$$

$$\Delta v = \frac{|v_1 - v_2|}{v_1 + v_2},$$

где T – темп записи, S_a – среднее значение спектра на всем диапазоне частот, S_l , S_m , S_h – средние значения спектра в диапазоне низких,

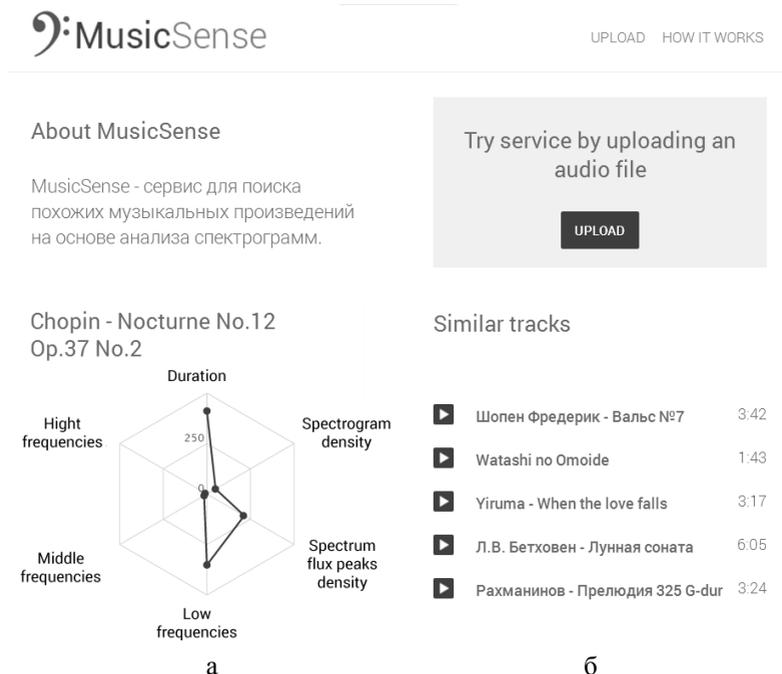
средних и высоких частот соответственно, L – длительность записи. Чем меньше значение D_{12} , тем более схожи композиции.

Направлением для дальнейшего совершенствования алгоритма может служить решение задачи определения жанра композиции и состава музыкальных инструментов, ее исполняющих.

4. Реализация алгоритма

Для проведения экспериментов была разработана тестовая платформа, реализованная в виде веб-приложения, состоящего из Java сервера, базы данных, Matlab-модуля и веб-интерфейса (см. рис. 2).

Веб-интерфейс предоставляет минимальный функционал для отправки аудиозаписи на сервер и отображения результатов работы сервиса. Результатами являются диаграмма (см. рис. 2а), отображающая вычисленные характеристики анализируемого трека и список похожих музыкальных произведений (см. рис. 2 б).



*а – диаграмма характеристик аудиозаписи,
б – список похожих произведений*

Рис. 2. Интерфейс веб-приложения

Сервер принимает загружаемые пользователем записи, анализирует их с помощью модуля, написанного на языке Matlab [2] и экспортированного в качестве библиотеки Java, затем ищет в базе данных наиболее похожие произведения и возвращает их на уровень веб-интерфейса. Анализируемая запись становится частью базы данных и участвует в качестве материала для поиска в последующих запросах.

5. Анализ качества рекомендаций

Для анализа оценки качества рекомендаций был проведен эксперимент. Были подготовлены две выборки: первая загружалась в базу и служила материалом для поиска, вторая выполняла роль контрольной. Выборка базы содержала 240 записей трех жанров: джаз, классика и рок (по 80 образцов). Контрольная выборка содержала по 20 образцов тех же жанров – 60 записей всего. Критерием успешной рекомендации считалось корректное определение жанра произведения.

В результате эксперимента система порекомендовала произведения того же жанра в 40 случаях, что составило 67% контрольной выборки. В большинстве случаев рекомендованные произведения были исполнены тем же составом инструментов, что и то, по которому осуществлялся поиск.

Заключение

Эксперимент показал, что созданный прототип алгоритма демонстрирует хорошие показатели качества рекомендации на контрольной выборке. Это подтвердило предположение о том, что подход поиска похожих музыкальных произведений, основанный на анализе аудиосигнала, может быть использован для повышения эффективности музыкальных рекомендательных систем.

Список литературы

1. Тим Джонс М. Рекомендательные системы [Электронный ресурс]. – Режим доступа : <https://www.ibm.com/developerworks/ru/library/os-recommender1/>
2. Документация MathWorks Matlab – The Language of Technical Computing [Электронный ресурс]. – Режим доступа : <http://www.mathworks.com/help/matlab/>
3. Adam H. Beat Detection for Automated Music Transcription: An exploration of Onset Detection Algorithms Computing [Электронный ресурс]. – Режим доступа : http://bingweb.binghamton.edu/~ahess2/Onset_Detection_Nov302011.pdf

АНАЛИЗ РИСКОВ ИСПОЛЬЗОВАНИЯ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ

М.М. Корчагин

Студент

С.А. Вялых

Доцент

Введение

Одной из важнейшей технологии для организации защищенного документооборота является электронная подпись. Электронная подпись позволяет идентифицировать владельца и служит гарантии целостности документа. Она обладает полной юридической силой согласно законодательству РФ. Как правило, с помощью электронной подписи производится шифрование документа при помощи ключа. Также электронная цифровая подпись дает возможность проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа. Несмотря на ряд преимуществ электронной цифровой подписи над рукописной, существуют вероятность кражи и утечки. Таким образом, использование электронной подписи несет свои риски.

1. Анализ наиболее важных факторов, влияющих на безопасность использования электронной подписи

Изготовление сертификата открытого ключа и проверки электронной подписи (СПЭП) осуществляется удостоверяющим центром (УЦ), на основании электронного запроса, при котором генерируется закрытый ключ (ЗК). Как правило, СПЭП и ЗК помещаются в специальный криптоконтейнер, доступ к которому осуществляется с использованием пин-кода. Наиболее распространённой сертифицированной технологией, используемой в настоящее время отечественными УЦ, является КриптоПро УЦ 1.5. Обычно генерация закрытого ключа и СПЭП осуществляется сотрудниками УЦ. Криптоконтейнер, как правило, формируется на защищённых носителях информации токенах или флэш-памяти.

СПЭП ключа на бумажном носителе, а также носитель информации, с записанными на него ЗК и СПЭП предоставляются его владельцу, или доверенному лицу, при личном обращении в УЦ.

Удостоверяющий центр – юридическое лицо или индивидуальный предприниматель, осуществляющие функции по созданию и выдаче сертификатов ключей проверки электронных подписей, а также иные функции, предусмотренные федеральным законом от 6.04.2011 № 63-ФЗ «Об электронной подписи» [1]. Типовой УЦ включает в себя следующие компоненты: центр сертификации, центр регистрации, автоматизированное рабочее место администратора, межсетевой экран, сервер публикации отозванных сертификатов пользователей УЦ. Через сеть Интернет с УЦ связаны рабочие места пользователей УЦ, включая лиц, проверяющих ЭП, которые имеют возможность в режиме реального времени проверять списки отозванных сертификатов, например, которые могли быть скомпрометированы на сервере публикации, и так же, в зависимости от регламента УЦ, иметь доступ к сервису центра регистрации. Далее представлена общая структура на рис. 1

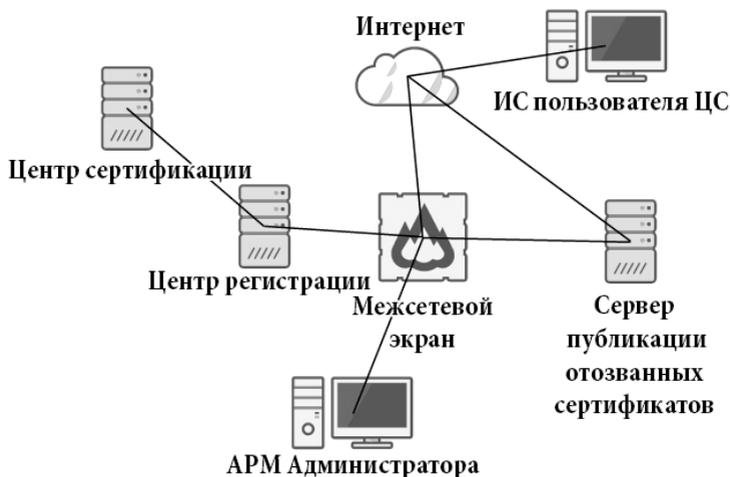


Рис. 1. Общая структурная схема УЦ вместе с рабочим местом пользователя

В настоящее время аккредитовано более 400 удостоверяющих центров. Их можно посмотреть на сайте e-trust.gosuslugi.ru/CA. Аккредитованный УЦ выпускает ЗК и СПЭП, которые используются для формирования квалифицированной ЭП. Информация в электронной форме, подписанная квалифицированной электронной подписью, признается электронным документом, равнозначным документу на бумажном носителе, подписанному собственноручной подписью [Федеральный закон от 6 апреля 2011 г. N 63-ФЗ].

Ключ генерируется удостоверяющим центром. Анализ рисков использования ЭП должен включать исследование возможности реализации угроз перехвата, или компрометации ЗК ЭП при его изготовлении или эксплуатации, при эксплуатации пользователем, а также угрозы возможности несанкционированного изготовления и использования ЭП пользователя. Для этого может использоваться любой из аккредитованных УЦ или скомпрометированный корневой ЗК одного из аккредитованных УЦ. Кроме этого необходимо оценить ущерб, который может быть нанесён пользователю.

В обобщенном виде угрозы безопасности информации характеризуется источниками угроз, факторами, обуславливающими возможность реализации угроз, способами (методами) реализации угроз и последствиями от реализации угроз безопасности информации [2]. Рассмотрим угрозы безопасности на трех главных этапах: генерация ЭП, получение ЭП, эксплуатации ЭП. На рис. 2 изображена модель угроз.

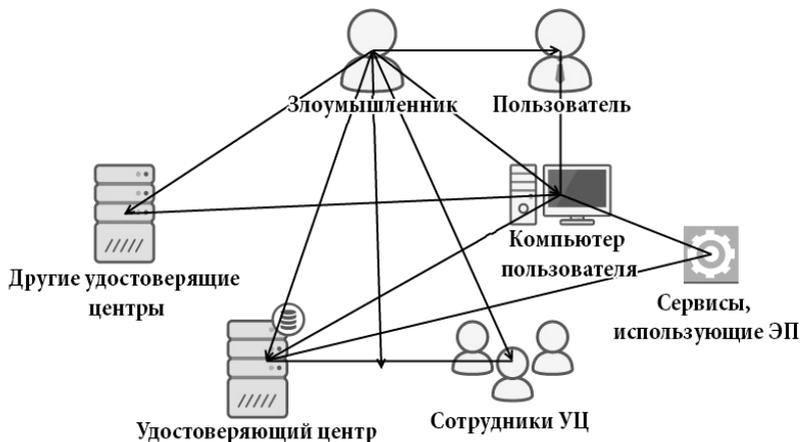


Рис. 2. Схема возможного перехвата данных.

Одним из наиболее уязвимых звеньев технологии ЭП является компьютер пользователя, который, как правило, не имеет достаточной защиты. Это позволяет реализовать следующие угрозы компрометации ЗК ЭП: перехват данных при записи в USB порт, копирование с носителя, подмена USB накопителя, запись на USB накопитель вредоносной программы. При обращении к ЗК возможно его несанкционированное использование для создания ЭП скрытых от пользователя поддельных документов, что не требует в ряде случаев, знания ЗК злоумышленником. Слабыми местами защиты компьютера пользователя являются:

- Отсутствие, или установка пиратской антивирусного приложения;
- Отсутствие контроля доступа посторонним лицам;
- Использование пиратского ПО.

Но даже если компьютер защищен, то при отсутствии знаний у пользователя, его можно легко обмануть, что так же приведет к утечке ЭП

Во время получения в УЦ ЗК и СПЭП, вместо пользователя может прийти злоумышленник с поддельными документами. Тем самым, ключ попадет не по назначению. Тут может быть два варианта: злоумышленник может украсть документы пользователя, и на их основе создать ЭП пользователя и скрыть от него этот факт. Выяснив, что его ЭП уже забрали, злоумышленник будет раскрыт, и ЭП будет заблокирована.

Во время эксплуатации ЭП ЗК, как указывалось выше, может находиться на двух типах носителей, это токены или флеш-память. Преимуществом токенов является двухфакторная аутентификация. Поэтому даже при краже или утери носителя ЗК, злоумышленнику будет сложнее воспользоваться ЭП. Флеш-память можно разделить на два основные группы, это USB-средства аутентификации и USB-флэш накопители. В USB-средства аутентификации защищены от чтения и записи. Пароль на доступ к носителю ЗК ЭП при его получении, часто может устанавливаться по умолчанию. При генерации ЗК на токен часто используются слабые, известные другим лицам, пароли, не отключают возможность копирования ЗК на другой носитель или используются не защищённые носители информации. Часто пользователь плохо разбирается в информационных технологиях и сам раскрывает пароль доступа другим лицам и даже передаёт носитель ЗК ЭП другим лицам для использования.

При использовании простого USB накопителя, злоумышленник без труда может скопировать информацию. Так же стоит учитывать тот факт, что при простом форматировании USB – накопителя, информацию можно восстановить. Кража или потеря одного из таких устройств хранения может быть легко замечена пользователем, после чего соответствующий сертификат может быть немедленно отозван [3].

Квалифицированную подпись может выпустить любой аккредитованный УЦ, без ведома пользователя. Этот тот случай, когда злоумышленник работает непосредственно с УЦ.

Таким образом можно сделать выводы, что утечка ЭП вполне возможна и реальна. Злоумышленник, зная слабые места, может с большой вероятностью украсть электронную подпись.

2.Оценка рисков

Электронная подпись очень часто используется в наше время. Взять к примеру сайт «Госуслуги», в котором пользователь может получить те или иные услуги. Если мы перейдем на сайт, то увидим несколько способов

авторизаций. Нас интересует способ авторизации через ЭП. Для авторизации необходимо скачать плагин.

Используя услуги, пользователь может не знать, что его ЭП была похищена. Злоумышленник может авторизоваться на портале «Госуслуги» без участия пользователя. Злоумышленник помимо получения новой услуги, сможет посмотреть и полученные услуги пользователя.

Многие услуги в случае несанкционированного использования ЭП злоумышленником могут рассматриваться как не несущие ущерба владельцу ЭП. Для примера оценки рассмотрим услугу «Судебная задолженность». Для оценки будем использовать методику оценки ФСТЭК. На рис. 3 изображен скриншот с сайта описания данной услуги.

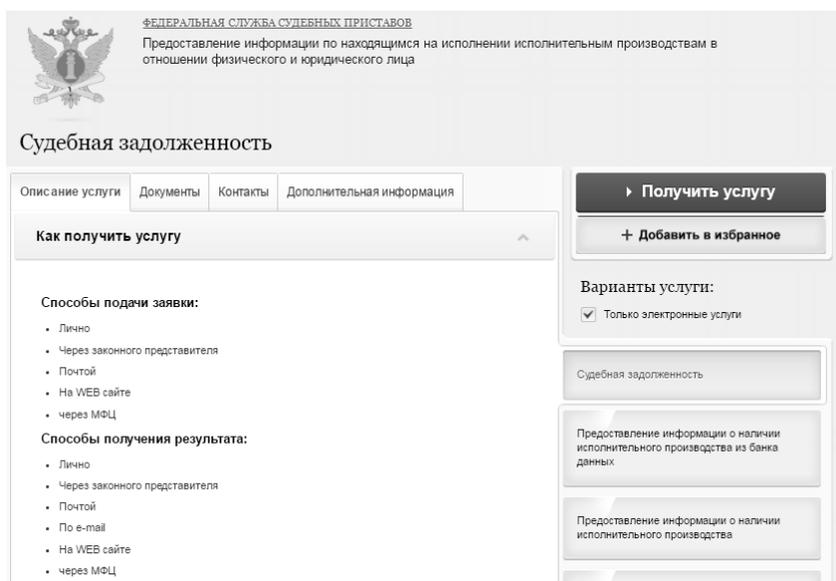


Рис. 3. Услуга на сайте

Документ ФСТЭК устанавливает единый методический подход к определению угроз безопасности информации и разработке моделей угроз безопасности информации в государственных информационных системах, защита информации в которых обеспечивается в соответствии с требованиями о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах [4].

Оценку рисков должен производить эксперт, который знает всю систему, и который знает, как утечка информации может так или иначе

повлиять на пользователя. В данном случае, выступая в роли эксперта произведем оценку.

Рассмотрев все риски по методике, и проанализировав информацию получаем следующий список рисков, которые могут быть реализованы при использовании данной услуги злоумышленником.

Экономические:

1. Потеря клиентов, поставщиков;
2. Невозможность заключение договоров, соглашений;
3. Другие прямые или косвенные финансовые потери.

Репутационный:

1. Нарушение деловой репутации;
2. Снижение престижа;
3. Утрата доверия.

Ущерб субъекту персональных данных:

4. Вторжение в частную жизнь;
5. Утрата репутации.

Исходя из этого, мы можем сделать выводы, что пользователь понесет ущерб в экономической, репутационной сфере. А также ущерб персональных данных.

Для более удобной оценки и сбора информации, был разработан сайт, который позволит оценить риски использования. Сайт предназначен для экспертов. На рис. 4 изображен итог оценки рисков использования электронной цифровой подписи.

Оценка

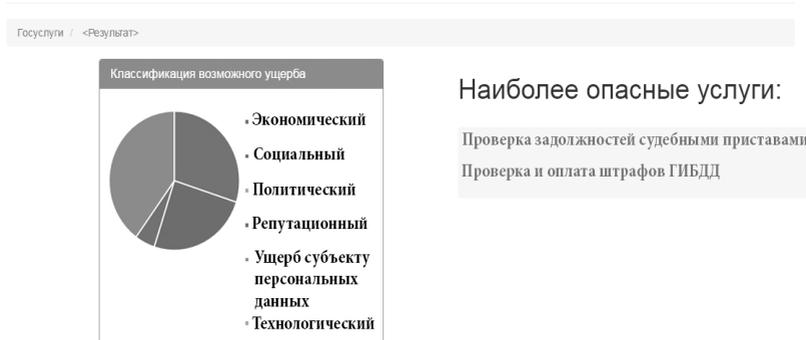


Рис. 4. Пример работы приложения

Мы можем добавлять удалять услуги, риски, менять методику оценки.

Так как информация хранится в базе данных, с этой информацией мы можем производить разные манипуляции. Например, если мы поменяем

формулу подсчета опасных услуг, то нам не придется производить оценку заново. Достаточно изменить скрипт сбора результатов.

Заключение

В данной статье проанализированы наиболее актуальные угрозы утечки электронной подписи. Для этого произведена оценка рисков услуги с сайта Госуслуг. Для более удобной оценки и сбора информации был разработан сайт, позволяющий вести статистику рисков для различных категорий. С помощью проведенных исследований, продемонстрировано, что использование услуг может быть крайне небезопасным. В данной работе рассмотрена только одна услуга, с сайта Госуслуг, но существуют и множество других порталов, где доступ предоставляется по электронной подписи. Подводя итоги проведенной работы отметим, что вероятность утечки информации на сегодняшний день довольно велика. Высокая степень ущерба может быть причинена не только пользователям, активно использующим ЭП в своей деятельности, но и любому гражданину. В связи с этим имеется необходимость в ранжировании информационных систем по степени возможного ущерба при неправомерном использовании ЭП, публикации данной информации в открытых источниках, и для систем, с возможной высокой степени ущерба, принятие мер адекватного противодействия.

Список литературы

1. Удостоверяющий центр [Электронный курс]. – Режим доступа : <http://www.gmcgks.ru>
2. Методика определения угроз безопасности информации в информационных системах [Электронный курс] : методика. – Режим доступа : <http://fstec.ru/component/attachments/download/812>
3. Гражданский кодекс [Электронный курс] : кодекс. – Режим доступа : <http://www.ville.ru/laws/gk/lawgk160.html>
4. Безопасность данных [Электронный курс]. – Режим доступа : <http://fstec.ru/component/attachments/download/289>

ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА ЛЕКАРСТВЕННЫХ ПРЕПАРАТОВ

В.П. Кузнецова

Студент

А.С. Миронова

Ассистент

Введение

Каждый человек имеет дело с лекарственными препаратами – временно или на постоянной основе, большинство инструкций которых достигают огромных размеров, а шрифт не позволяет прочесть их даже здоровому и хорошо видящему человеку. Кроме того, чтобы найти интересующую в данный момент информацию о препарате, необходимо изучить инструкцию вдоль и поперек, потратив немало времени в поисках нужного раздела. Ко всему прочему, существует множество лекарственных средств, которые требуют жесткого расписания приема и пациент зачастую не в состоянии контролировать это должным образом.

С другой стороны, мобильные портативные устройства вошли в нашу жизнь окончательно и бесповоротно. На данный момент большинство населения пользуется смартфонами, которые предоставляют возможность установить множество приложений, облегчающих жизнь.

Именно по этим причинам в данной статье рассматривается проблема построения системы, которая должна решать перечисленные выше проблемы и предоставлять хранение информации о лекарственных препаратах и ее отображение в удобном, структурированном виде, с целью быстрого доступа к интересующему разделу инструкции, и создания напоминания о приеме лекарственного средства.

1. Постановка задачи

Необходимо разработать систему для поиска лекарственных средств и отображения инструкции препаратов в удобном виде. Данная система должна удовлетворять следующим требованиям:

1. Поиск лекарственного средства должен осуществляться различными способами:
 - ввод названия;
 - сканирование штрих-кода;
 - распознавание названия по фото упаковки.
2. Представление инструкции в удобном, структурированном виде с возможностью быстрого доступа к определенному разделу.

3. Добавление препарата в избранное.
4. Сохранение истории поиска.
5. Добавление напоминания принять лекарство, предполагающее создание расписания приема средства.
6. Возможность поиска и просмотра наиболее используемых средств без подключения к глобальной сети.
7. Эргономичность и простота пользовательского интерфейса приложения. Необходимо построить графический интерфейс, удобный для использования людьми с нарушениями зрения.

2. Анализ предметной области

Прежде всего, был проведен анализ существующих аналогов, на основе которого были сделаны следующие выводы:

- ни одно приложение-аналог не отображает инструкцию препарата в структурированном виде;
- практически все аналоги используют Интернет-соединение для поиска инструкции;
- ни одно приложение не предоставляет возможность напоминания принять препарат;
- все приложения осуществляют поиск только через непосредственный ввод наименования лекарства.

Таким образом, ни одно приложение-аналог не способно в полной мере удовлетворить пользовательские потребности.

Для обеспечения соответствия системы представленным требованиям были рассмотрены и проанализированы потенциальные проблемы, которые могли возникнуть при разработке системы.

На рис. 1 представлены возможные варианты использования серверной части разрабатываемой системы:

1. Обработка запроса пользователя представляет собой выборку из базы данных, сформированную по требованию пользователя. По полученному от пользователя названию препарата формируется SQL-запрос, с помощью которого сервер обращается к базе данных, осуществляя поиск нужного средства.
2. Обработка полученных данных подразумевает формирование на основе полученной из базы данных выборки ответа в понятном для клиентского приложения формате и его последующую отправку.

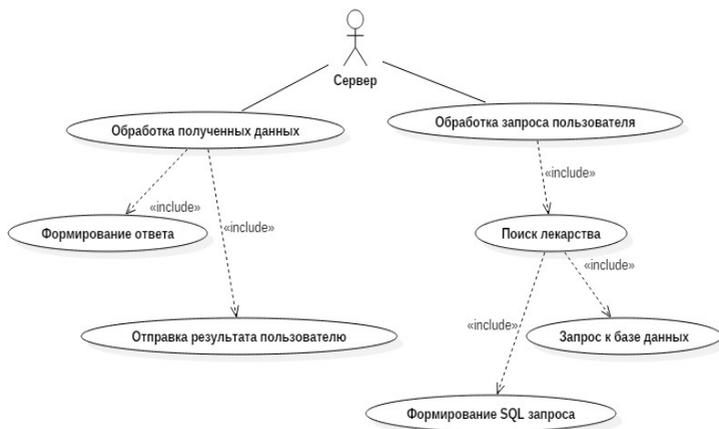


Рис. 1. Диаграмма вариантов использования серверной части системы

Так как серверная часть системы оперирует данными, хранящимися в базе данных всех лекарств и содержащими информацию о препаратах, на основе ER-модели предметной области (рис. 2) была выделена основная сущность – «Инструкция». «Инструкция» полностью описывает лекарственный препарат и содержит в себе множество полей, часть из которых связана с сущностями «Лекарственная форма», «Фармакотерапевтическая группа» и «Условия отпуска из аптек». Помимо обычных текстовых полей, «Инструкция» содержит в себе данные составного характера, такие как «Фармакокинетика» и «Показания к применению», которые у каждого препарата уникальны, причем не каждое средство имеет данные о фармакокинетике. На основе данной модели была спроектирована реляционная база данных [1].

В качестве архитектуры проектируемой системы выбрана трехуровневая архитектура клиент-серверных систем (рис. 3) по причине ее переносимости, модульности, надежности, простоты в интеграции. Серверное приложение предоставляет разработчику клиентской части API, в котором реализован доступ к ресурсам системы. В качестве модели реализации API была выбрана модель REST [2] в силу ее простоты архитектурного решения и ориентированности на работу с данными.

Таким образом, к задачам API проектируемой системы относятся:

- обработка полученного от пользователя запроса на поиск конкретного лекарственного препарата;
- обработка полученного результата поиска средства.

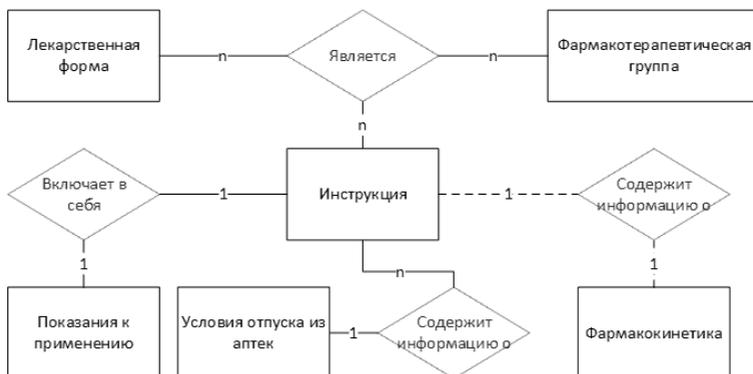


Рис. 2. ER-модель предметной области серверной части системы



Рис. 3. Трёхуровневая архитектура клиент-серверной системы

Клиентская часть проектируемой системы представляет собой приложение, диаграмма вариантов использования которой (рис. 4) включает в себя следующие функции, которыми должно обладать клиентское приложение:

1. Поиск интересующего лекарственного препарата. Данное действие пользователь может осуществить несколькими способами.
2. Просмотр инструкции лекарственного средства с возможностью навигации по ее разделам.
3. Возможность управлять списком избранных лекарственных препаратов.
4. Создание напоминания о приеме необходимого средства.

Опираясь на статистику [3, 4] за ноябрь – декабрь 2015 года, в качестве устройства-носителя клиентского приложения решено использовать мобильное устройство под управлением ОС Android шестнадцатой версии API и выше.

На основе анализа данных, с которыми должно оперировать клиентское приложение, и способов их хранения в устройствах на базе ОС Android было решено:

- для хранения списка избранного, заметок и приемов организовать базу данных SQLite;
- для хранения недавно найденных препаратов использовать кэш приложения, чтобы обеспечить быстрый доступ к данным.

Кроме того, был проведен анализ пользовательского интерфейса, в ходе которого были спроектированы экраны клиентского приложения.

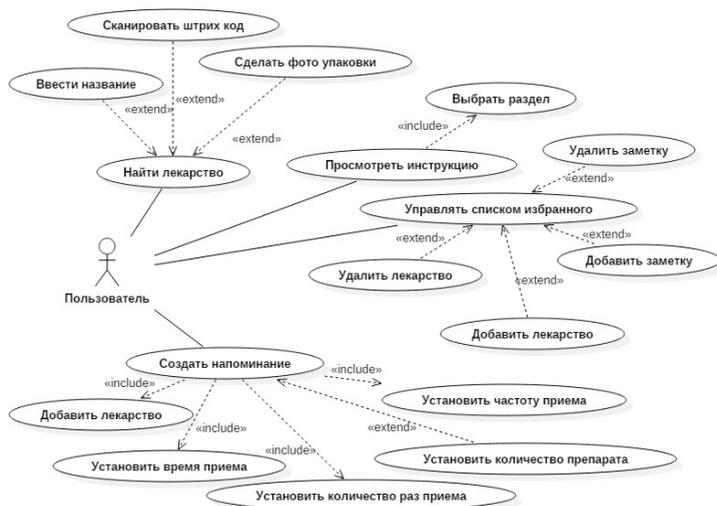


Рис. 4. Диаграмма вариантов использования клиентской части системы

3. Реализация

Структурная схема, представленная на рис. 5, иллюстрирует, что система состоит из двух основных частей:

- Server – представляет собой серверную часть;
- Client – является клиентским приложением, установленном на мобильном портативном устройстве.

Серверная часть системы основана на паттерне проектирования Model-View-Controller (MVC) [5] и делится на следующие модули:

1. Database – база данных, отвечающая за хранение информации о лекарственных препаратах.
2. Model – модуль, представляющий собой объектную модель системы на основе сущностей базы данных.
3. Controller – модуль, состоящий из следующих частей:
 - API – модуль, предоставляющий интерфейс взаимодействия с мобильным клиентом;
 - Web-logic – модуль, реализующий логику организации поискового алгоритма.
4. View – модуль отображения данных в браузере на основании шаблонов формирования веб-страниц (Templates).

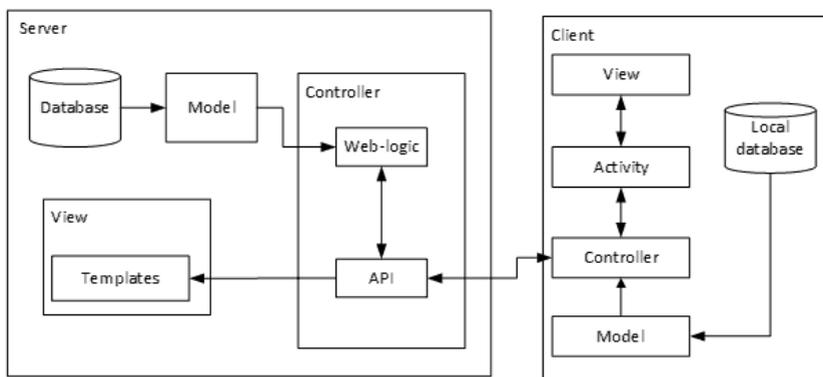


Рис. 5. Структурная схема системы

Мобильный клиент так же основан на паттерне проектирования MVC и подразделяется на следующие модули:

1. Local database – база данных, отвечающая за хранение информации об избранных лекарственных препаратах, заметках и приемах пользователя в пределах локального хранилища ОС Android.
2. Model – модуль, представляющий собой объектную модель клиентской части системы на основе сущностей локальной базы данных.
3. Controller – модуль, реализующий взаимодействие с сервером.
4. Activity – модуль, специфичный для платформы Android. Является посредником между модулями View и Controller.
5. View – модуль, отвечающий за представление пользовательского интерфейса мобильного клиента.

Заключение

В данной статье описаны основные этапы проектирования и реализации информационно-поисковой системы лекарственных препаратов. В результате проделанной работы была получена полностью работающая система, предоставляющая поиск лекарства различными способами, отображающая инструкции средств в структурированном, удобном виде, сохраняющая дополнительную пользовательскую информацию, такую как список избранных препаратов и заметки, позволяющая создавать оповещения о приемах лекарств.

Список литературы

1. Харрингтон Дж. Л. Проектирование реляционных баз данных / Дж. Л. Харрингтон. – М. : Лори, 2006. – 230 с.

2. Richardson L. RESTful Web Services / L. Richardson, S. Ruby. – Sebastopol : O'Reilly Media, 2007. – 454 с.
3. LiveInternet [Электронный ресурс] : сервис статистики Интернет ресурсов. – Режим доступа: <http://www.liveinternet.ru>
- Google Developers Dashboards [Электронный ресурс] : сервис статистики пользовательских устройств от Google. – Режим доступа : <http://developer.android.com/intl/ru/about/dashboards/index.html>
4. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма [и др.]. – СПб. : Питер, 2001. – 368 с.

МОНИТОРИНГ ЦЕЛОСТНОСТИ ВЕБ-САЙТА

А.Г. Ласкин

Студент

С.А. Вялых

Доцент

Введение

Информационные ресурсы являются объектом права собственности и не менее ценным ресурсом, чем материальные или энергические. Поэтому их защита признана сегодня наиболее актуальной проблемой. При этом защита целостности это один из главных аспектов обеспечения безопасности информационных ресурсов. Под целостностью информации понимается такое состояние информации, при котором в ней отсутствуют ненадлежащие изменения после выполнения каких-либо операций (передача, хранение или отображение информации) [1].

Одними из наиболее популярных информационных ресурсов сейчас являются веб-сайты. Около 60% всех веб-сайтов хотя бы раз подвергались хакерской атаке [2]. Чаще всего взломанные сайты используются для размещения на них перенаправляющих скриптов и вредоносных программ. Поэтому любой владелец сайта заинтересован в том, чтобы обеспечить его безопасность, в частности целостность веб-сайта.

1. Анализ угроз целостности веб-сайтов

Цель работы, которая рассматривается в рамках данной статьи, – анализ угроз целостности веб-сайтов и разработка средства мониторинга целостности веб-сайта.

При изучении вопроса обеспечения целостности веб-сайтов можно выделить следующие наиболее распространенные угрозы:

– Угроза SQL инъекции – вставка в форму специальным образом сформированного выражения, которое будет выполнено при запросе к базе данных (позволяет обойти парольную защиту, украсть конфиденциальные данные или уничтожить их и др.);

– Угроза XSS инъекции – вставка кода, который будет выполнен в браузере посетителя веб-сайта (чаще всего таким способом крадут файлы cookie с целью получения доступа от имени посетителя, а также перенаправляют пользователей на другие ресурсы: платная реклама, зараженные сайты, фишинговые сайты);

– Угроза взлома или компрометации ssh и ftp паролей с целью изменения файлов сайта и размещения вредоносного кода.

Первые две проблемы решаются проверкой всех данных, получаемых скриптами веб-сайта в POST и GET запросах, а так же удалением запрещенных символов или конструкций и, при необходимости, экранированием данных. А вот от третьей угрозы, к сожалению, защититься сложнее.

По статистике веб-проекта Zone-H за 2010 год [3] наиболее распространенными уязвимостями в веб-скриптах являлись SQL-инъекции (98 250 случаев), а также возможность загрузки удаленных файлов и внедрения вредоносного программного кода (634 620 случаев).

Кроме того, по данным Cisco [4] в период с февраля по октябрь 2015 года число используемых преступниками доменов ныне популярной платформы WordPress стремительно возросло на 221 %.

Как отмечают в Cisco, взломанные сайты WordPress часто работали не на последней версии WordPress, отличались слабыми паролями администратора и использовали подключаемые модули без исправлений информационной безопасности.

В ходе исследования, проведенного Cisco в октябре 2015 г. [4], было выявлено, как ИТ-руководители воспринимают риски и проблемы информационной безопасности. Таким образом, 34% участников опроса назвали внедрение вредоносного ПО основной внешней проблемой информационной безопасности организации. Фишинг и сложные целенаправленные угрозы заняли второе и третье место — 27% и 21% соответственно.

2. Анализ уязвимостей веб-сайта

«Информационная безопасность Воронежской области»

Для практического изучения уязвимостей веб-сайтов был выбран проект сайта правительства «Информационная безопасность Воронежской области». В результате его проверки через систему WordPress Security Scanner (рис. 1) было выявлено наличие нескольких уязвимых для атак установленных плагинов и устаревшая версия WordPress.

Для минимизации возможностей атак с помощью данных уязвимостей программой было предложено обновить эти плагины до последней версии. Однако в любом случае веб-сайт всё еще остается подвержен несанкционированному доступу, что демонстрирует актуальность проблемы защиты целостности веб-сайтов.

Analysis of <http://ibvrn.16mb.com>

	WordPress Version 4.1.10 Version does not appear to be latest 4.5 - update now.	Google safe browse check PASSED Google finds the site to be free of malware	Server: Apache X-Powered-By: PHP/5.5.34 IP Address: 31.170.165.198 Provider: Hostinger International Limited Country: United Kingdom
--	---	---	---

WordPress Plugins

The following plugins were detected by reading the HTML source of the WordPress sites front page.

Plugin Name	Latest Release	Update Required
wp-ds-blog-map 3.1.3	latest release (3.1.3) http://wp-plugins.diamondsteel.ru/	No
wordpress-popular-posts 3.2.1	latest release (3.3.3) http://wordpress.org/extend/plugins/wordpress-popular-posts	Yes
ajax-search-lite 4.5.1	latest release (4.6.2) http://wp-dreams.com	Yes
wp-glossary 3.1.1.2	latest release (3.1.1.2) http://wordpress.org/extend/plugins/wp-glossary/	No
bbpress 2.5.8	latest release (2.5.8) http://bbpress.org	No
wp-pagenavi 2.87	latest release (2.90) http://lesterchan.net/portfolio/programming/php/	Yes

Рис. 1. Анализ уязвимостей веб-сайта через систему WordPress Security Scanner

3. Средство мониторинга целостности веб-сайта

На основе результатов, полученных в ходе анализа актуальных угроз целостности и уязвимостей веб-сайтов, средствами языка PHP для сайта «Информационная безопасность Воронежской области» была разработана программа по поддержке целостности веб-сайта, направленная на мониторинг целостности файлов и скриптов сайта и на своевременное оповещение владельца веб-сайта в случае нарушения их целостности.

Полная функциональность средства включает в себя:

- Проверку целостности файлов веб-сайта в двух возможных режимах: по размеру и дате изменения или значению md5-хэша;
- Оповещение пользователя по e-mail в случае наличия изменений в файлах веб-сайта;
- Указание исключений для проверки в виде директорий, конкретных файлов или типов файлов;

– Возможность систематического запуска с помощью Cron или любого другого планировщика, работающего с php-файлами.

Тестирование разработанного средства мониторинга целостности веб-сайта проводилось при помощи портативной серверной платформы Open Server 5.2.2 на примере одного из собственных сайтов.

Перед запуском программы ее исходные файлы помещаются на сервер в директорию с файлами сайта. В конфигурационном файле `site_scanner_config.ini` указываются e-mail, на который будут приходить уведомления в случае нарушения целостности файлов веб-сайта, и режим работы сканирования: 0 – проверка файлов по размеру и дате изменения, 1 – проверка файлов по значению md5-хэша. Второй режим является более надежным, но требует больших ресурсных затрат при большом объеме проверяемых файлов.

База с информацией о файлах сайта хранится в файле `site_data.json` и обновляется после каждого сканирования. База с исключениями для сканирования в виде названий директорий, файлов и типов файлов хранится в файле `site_exemption_list.json`.

В результате запуска программы `site_scanner.php` через браузер в окне браузера выводится информация об обнаруженных нарушениях целостности веб-сайта с момента последнего сканирования (рис. 2).

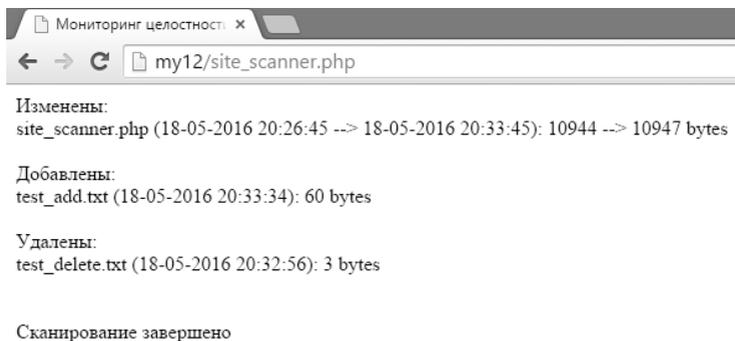


Рис. 2. Вывод результата сканирования веб-сайта в браузере

Также при наличии нарушений целостности веб-сайта на указанный в конфигурационном файле e-mail высылается уведомление (рис. 3), содержащее информацию об изменениях в файлах веб-сайта.



Изменения на сервере my12

██████████@gmail.com ↕

Кому: ██████████@mail.ru
сегодня, 20:33

С момента последнего сканирования (18-05-2016 20:33:02) директорииD:/OpenServer/domains/my12/ на сервере my12 произошли изменения. Файлов просканировано: 23,
Обнаружено измененных файлов: 1,
Обнаружено добавленных файлов: 1,
Обнаружено удаленных файлов: 1

Изменены:

site_scanner.php (18-05-2016 20:26:45 --> 18-05-2016 20:33:45): 10944 --> 10947 bytes

Добавлены:

test_add.txt (18-05-2016 20:33:34): 60 bytes

Удалены:

test_delete.txt (18-05-2016 20:32:56): 3 bytes

Рис. 3. Уведомление по e-mail о нарушении целостности веб-сайта

Процесс сканирования файлов веб-сайта сопровождается логированием в файл site_log.log (рис. 4).

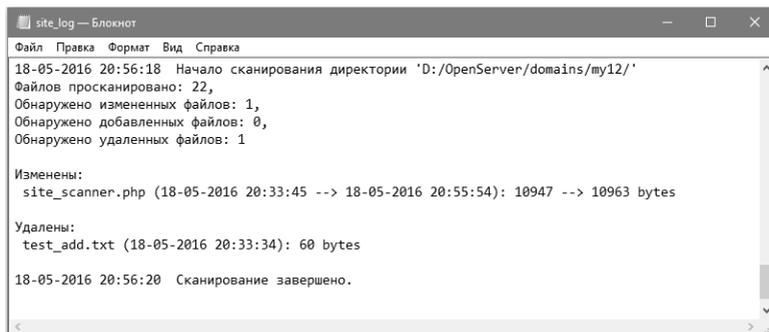


Рис. 4. Пример результата сканирования в логе

Для автоматического периодического запуска программы можно использовать Стон или любой другой планировщик задач, работающий с php-файлами.

В качестве дальнейших перспектив развития проекта рассматривается возможность добавления оповещения владельца веб-сайта о нарушениях целостности по sms, восстановления файлов после изменения или удаления, а также разработка графического интерфейса взаимодействия с программой.

Заключение

В данной статье проанализированы наиболее актуальные угрозы целостности веб-сайтов, и описано разработанное на основе этого анализа

средство мониторинга целостности веб-сайта для проекта сайта правительства «Информационная безопасность Воронежской области». Данное средство позволяет отслеживать любые изменения в файлах веб-сайта и своевременно оповещать о них владельца по электронной почте, что снижает риск распространения вредоносного кода в случае его внедрения и уменьшает ущерб от его деятельности.

Список литературы

1. Грибунин В. Г. Комплексная система защиты информации на предприятии : учеб. пособие для студ. высш. учеб. заведений / В. Г. Грибунин, В. В. Чудовский. – М. : Издательский центр «Академия», 2009. – 416 с.
2. Петренко С. А. Политики информационной безопасности / С. А. Петренко, В. А. Курбатов – М. : ДМК Пресс, 2010. – 400 с.
3. Defacements Statistics 2010 [Электронный ресурс] : база данных. – Режим доступа : <http://www.zone-h.org/news/id/4737>
4. Годовой отчет Cisco по информационной безопасности за 2016 год [Электронный ресурс] : электр. журн. – Режим доступа : <http://www.slideshare.net/CiscoRu/cisco-2016>

УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ПО РАСПИСАНИЮ

П.О. Левчук
Студент
С.А. Зуев
Ассистент

Введение

В настоящее время, в целях повышения уровня безопасности, а так же для усовершенствования качественных характеристик произведенных товаров, ученые стремятся максимально отделить процесс производства от возможного негативного влияния человеческого фактора. Для осуществления данных целей в настоящий момент чаще всего используют автоматизированные системы управления. Такие системы в зависимости от своих ключевых характеристик делятся на два вида:

6. Временные системы (позволяют управлять длительностью процессов, задают время его начала и окончания);

7. Параметрические системы (делают возможным управление технологическими процессами в соответствии с заданным алгоритмом управления, а также измеряемыми и задаваемыми параметрами).

Наиболее перспективным методом автоматизированного управления сложными динамическими системами (процессами) является диспетчерское управление и сбор данных (SCADA Supervisory Control And Data Acquisition). Именно на принципах диспетчерского управления строятся современные масштабные автоматизированные системы в промышленности и энергетике, транспорте, космической и военной областях, различных государственных структурах. Суть данного метода заключается в процессе сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами. Возникающая потребность обработки именно реального времени обусловлена, в первую очередь, необходимостью доставки (выдачи) всех необходимых событий (сообщений) и данных на центральный интерфейс оператора (диспетчера).

1. SCADA-системы

Существует несколько разновидностей архитектуры SCADA-систем:

1. Одиночные: При использовании данной архитектуры система состоит из одной или нескольких рабочих станций оператора, которые не "знают" друг о друге. Все функции системы выполняются на единственной (нескольких независимых) станции(ях). Преимущества: простота, низкая отказоустойчивость не обеспечивается истинность данных (исторические данные могут отличаться между разными станциями)

2. Клиент-серверные: В данном случае система выполняется на сервере, а операторы используют клиентские станции для мониторинга и управления процессом. Высоконадёжные системы строятся на базе двойного либо тройного резервирования серверов и дублирования клиентских станций оператора, дублирования сетевых подключений сервер-сервер и клиент-сервер. При данной архитектуре уже возможно разделение функций SCADA-системы между серверами. Например, сбор данных и управление ПЛК выполняется на одном сервере, архивирование данных - на втором, а взаимодействие с клиентами - на третьем.

3. Виртуализация: Современные технологии виртуализации уже глубоко проникли в промышленную автоматизацию. Применение кластеров и виртуальных серверов с разделением функций SCADA-системы позволяет обеспечивать высокую отказоустойчивость, гибкое распределение вычислительных ресурсов, изолирование системы и сетевых подключений от постороннего сетевого трафика, безопасность данных. При использовании виртуализации клиентские станции уже не

требуют полноценного ПК(Толстый клиент), достаточно тонкого клиента с подключением к виртуальному клиенту. Облачные вычисления также применяются в промышленной автоматизации либо автоматизации зданий.

Однако, несмотря на передовой характер системы диспетчерского управления и сбора данных, как наиболее часто используемой системы, данный метод также не лишен определенных недостатков. Главным из которых, является финансовая нецелесообразность: объем общих затрат, необходимых для обеспечения функционирования подобной системы, не соотносится с количеством получаемых в итоге функциональных преимуществ от ее использования [1].

2. Постановка задачи

Необходимо создать действующий макет системы управления технологическими процессами в реальном масштабе времени на современной программно-аппаратной базе.

В данной статье проведён анализ возможных компонентов автоматизированных систем и создание своей системы для управления технологическими процессами по расписанию при минимальных затратах.

Осуществление поставленной цели потребовало проведения тестирования, для которого мы выбрали аппаратную платформу IBM PC и операционную систему Windows (начиная с Windows XP и более поздние редакции — Windows7, 8 и 10), систему синхронизации времени NTP и плату USB-контроллера с восемью реле в одном модуле — PL2303.

3. Системы синхронизации времени

Одним из основных компонентов автоматизированных систем по расписанию является система синхронизации времени, которая используется для обеспечения временной точности автоматизации и отказоустойчивости управляемого оборудования. На современном этапе развития разработчики используют следующие системы синхронизации времени:

4. NTP – сетевой протокол для синхронизации внутренних часов компьютера, использующий UDP и учитывающий время передачи сообщения, в связи с чем, зависимость от его изменения не критична. Данная система способна достигать точности 10 мс при работе через Интернет или 0.2 мс при функционировании в локальной сети.[2]

5. Радиоканалы, которые формируют точное время на основании показаний атомных часов. Атомные часы - прибор для измерения времени, в котором в качестве периодического процесса используются колебания в ядре атома вокруг электронов, за счёт чего погрешности крайне малы. Наиболее точный вариант, основанный на Цезии-133, определяет секунду как 9192631770 периодов электромагнитного излучения, возникающего

при переходе между двумя сверхтонкими уровнями основного состояния атома [3].

4. Описание программы.

Для исследования была разработана программа управления устройствами, подключёнными к плате PL2303, на языке C#. Данная программа позволяет управлять USB модулем и своевременно подавать напряжение на отдельные реле, подключённые к ней. Для управления платой, которая читается PC как COM порт, используется встроенный класс Serial Port с методами Open, Write, Read и Close, позволяющими открывать и закрывать поток USB платы, а также отправлять или принимать байты информации. Используя класс Timer из библиотеки System.Timers с периодичностью 1мс, вызывается проверка соответствия между текущим временем системы и временем перехода платы в другое состояние [4]. При установлении подобного соответствия, вызывается событие **TimerExecuted**, на которое реагирует главное окно программы, и поочередно осуществляются следующие действия:

1. проверка подключения устройства к PC;
2. открытие потока, если он не открыт;
3. отсылка байта памяти, передающегося из файла Settings.txt с помощью класса TimerExecutedArgs как аргумент события;
4. закрытие потока, если он не был открыт.

В файле **Settings.txt** хранится массив класса **Alarm**, содержащий в себе поля типов **DateTime** и **byte**, представляющих время исполнения и отсылаемую команду [5]. Рис. 1 поясняет алгоритм работы программы.

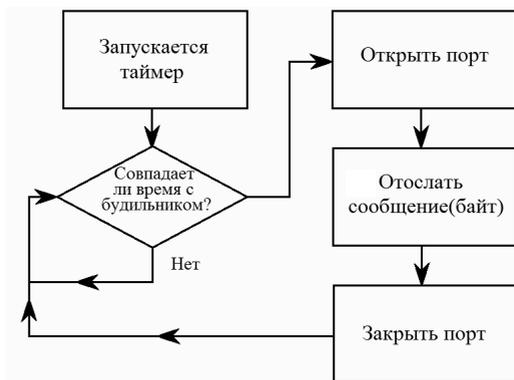


Рис. 1. Алгоритм работы программы

На рис. 2-4 приведены примеры управления работой программы.

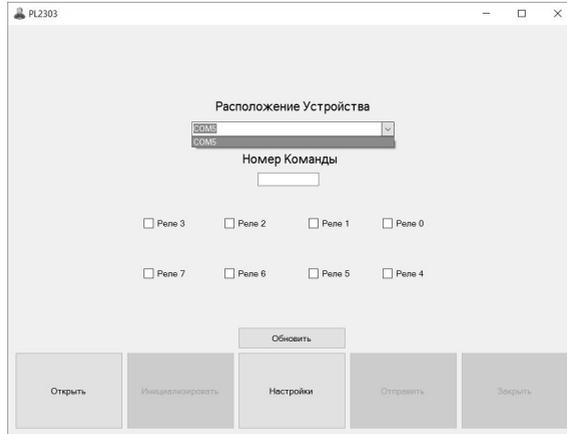


Рис. 2. Выбор номера СОМ-порта.

В данном окне программы мы можем:

- 1) Выбрать разъём, в котором находится наше устройство
- 2) Открыть порт, чтобы плата могла принимать сообщения от программы.
- 3) Инициализировать плату, послав ей байты 78 79 80 81, для того, чтобы она могла реагировать на посланные сообщения
- 4) Заккрыть порт.
- 5) Перейти в меню настроек, для установки команд посылаемых плате в определённое время.

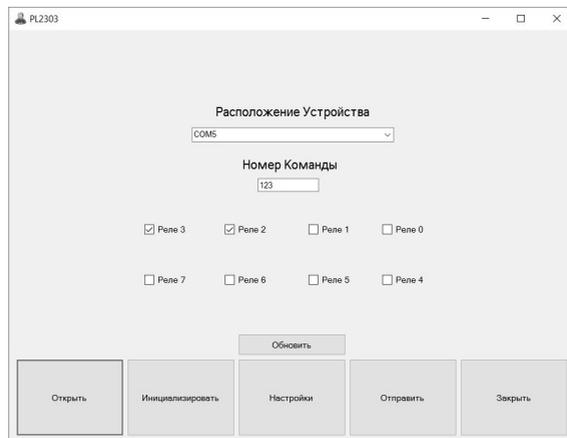


Рис. 3. Задание номеров задействованных реле

В данном окне программы отображаются все заданные пользователем будильники. Пользователь может совершить данные действия:

- 1) Создать новый будильник.
- 2) Удалить все будильники.
- 3) Сохранить изменения.
- 4) Отменить изменения.

Основываясь на этих функциях описываемой программы, было создано приложение для управления звонками в школе или любом другом учебном заведении по расписанию. При этом применённый модуль USB-реле является явно избыточным: для управления электрическими звонками в школе достаточно одного реле. Гибкость настроек расписания позволяет создать систему, настраиваемую один раз в году, с учётом праздников и выходных дней, а также особых событий в жизни данного учреждения. При этом в нерабочие дни система полностью отключает схему питания звонков во всём здании.

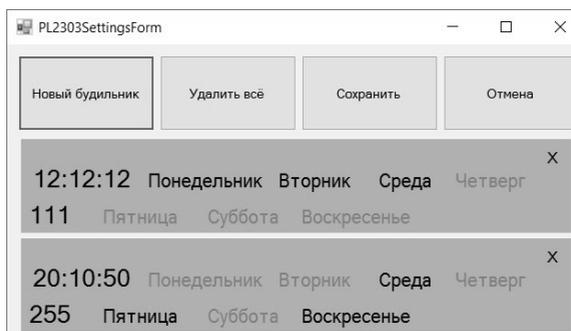


Рис. 4. Создание расписания срабатывания реле.

В данном окне программы задаётся новый будильник. Полный листинг программы приведён в [6].

Заключение

Статья посвящена системам управления технологическими процессами по расписанию. Описано создание такой системы, устройства для её реализации и их программирование. Приведён практический пример приложения для подачи звонков по расписанию. В итоге создана полностью рабочая система управления технологическими процессами по расписанию. Данное решение позволяет управлять 256 устройствами (при наличии дешифратора) с минимальными денежными затратами.

Источники

1. Статья из Википедии [Электронный ресурс]. — Режим доступа : <https://ru.wikipedia.org/wiki/SCADA>
2. Статья из Википедии [Электронный ресурс]. — Режим доступа : <https://ru.wikipedia.org/wiki/NTP>
3. Статья из Википедии [Электронный ресурс]. — Режим доступа : https://ru.wikipedia.org/wiki/Атомные_часы
4. Библиотека MSDN Microsoft [Электронный ресурс]. — Режим доступа : [https://msdn.microsoft.com/ru-ru/library/system.io.ports.serialport\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.io.ports.serialport(v=vs.110).aspx)
5. Библиотека MSDN Microsoft [Электронный ресурс]. — Режим доступа : [https://msdn.microsoft.com/ru-ru/library/system.datetime\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.datetime(v=vs.110).aspx)
6. Электронное хранилище Google [Электронный ресурс]. — Режим доступа : <https://docs.google.com/document/d/1yZ0u10-xGWGb72vGP-PH4BAoIGMZ2A7XDDULwhZqtKg/edit#heading=h.1t3h5sf>

РЕАЛИЗАЦИЯ СЕНСОРНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ МИКРОКОНТРОЛЛЕРОВ MSP430

Р.В. Литвинов
Студент
Д.Н. Борисов
Доцент

Введение

В наше время беспроводная связь выходит на первый план. По сравнению с проводной связью, она имеет ряд преимуществ: построение сети в местах, где затруднено реализовать проводную связь, отсутствие проводов и кабелей, нет необходимости в проведении монтажных работ и самое главное – мобильность и масштабируемость. Беспроводная сенсорная сеть – сеть, состоящая из множества датчиков и устройств управления (узлы сети). Совокупность таких узлов образует самоорганизующуюся и распределенную в пространстве систему сбора, обработки и передачи информации для ее дальнейшего использования или анализа. Существуют различные топологии построения беспроводной сенсорной сети, однако самой эффективной является ячеистая топология (mesh-сеть). Каждый узел в mesh-сети способен ретранслировать пакеты данных на другой узел, тем самым увеличивая область покрытия сети и ее

отказоустойчивость. Микроконтроллеры семейства MSP430 в совокупности с радиочастотным трансивером CC2500 отлично подходят в качестве узла mesh-сети. Данные микроконтроллеры просты в освоении и имеют сверхнизкое энергопотребление, что является одним из ключевых факторов при построении беспроводной сенсорной сети.

Для построения беспроводной mesh-сети, используя радиочастотный трансивер CC2500, необходима программная реализация протокола AODV. С его помощью реализуется динамическая маршрутизация между узлами сети. Стандартные библиотеки фирмы производителя Texas Instrument не позволяют реализовать mesh-сеть, используя трансивер CC2500. Результат данного исследования – программное обеспечение, которое позволит строить беспроводные сенсорные сети с ячеистой топологией, используя в качестве узла микроконтроллер семейства MSP430 со сверхнизким энергопотреблением и трансиверы CC1100, CC2500.

1. Микроконтроллеры семейства MSP430

MSP430 – семейство микроконтроллеров фирмы Texas Instruments (TI), обладающих сверхнизким энергопотреблением и предназначенных для обработки смешанных (цифровых и аналоговых) сигналов. Поддержка различных режимов энергосбережения предоставляют разработчику мощный инструмент для разработки приложений, в которых требуется беспроводной прием и передача данных другим устройствам. В таком случае микроконтроллеры семейства MSP430 будут находиться в спящем режиме, до тех пор, пока не потребуется совершить какое-либо действие. При реализации сенсорных сетей, требующих длительное время работы устройства без замены элемента питания, использование подобных режимов позволяет разрабатывать наиболее эффективное программное обеспечение для работы с беспроводными сенсорными сетями. В процессе исследования используется плата RF2500 фирмы TI, включающая в себя микроконтроллер MSP430F2274, трансивер CC2500, датчик температуры и различные периферийные устройства (рис. 1).

Главными преимуществами микроконтроллеров семейства MSP430 являются:

- Поддержка различных режимов энергосбережения [1];
- Обработка смешанных (аналоговых и цифровых сигналов);
- Использование современных технологий программирования.

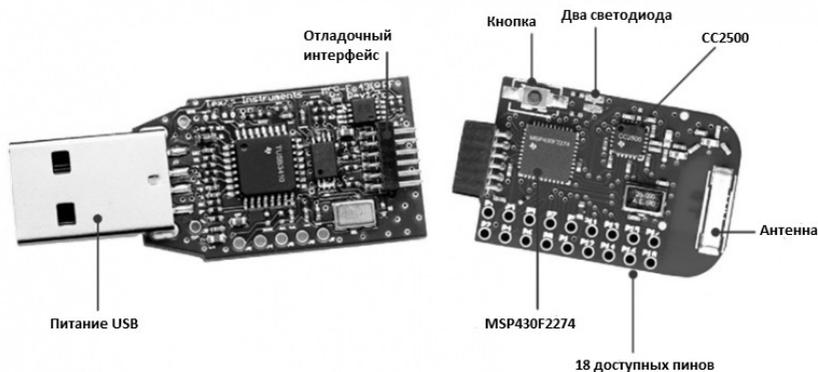


Рис. 1. Плата RF2500

2. Среда разработки Code Composer Studio

Для программирования микроконтроллеров семейства MSP430 используется среда разработки Code Composer Studio, предоставляемая фирмой производителем TI. Данная среда проста в освоении, обладает широким набором функций, необходимых в процессе разработки программного обеспечения на базе микроконтроллеров MSP430, и удобным графическим интерфейсом. Программирование устройства осуществляется на языке Си.

Среда включает в себя редактор исходных текстов, отладчик, линковщик, компилятор, средства визуализации, симулятор любого встроенного процессора от TI, а также целый набор вспомогательных инструментов. Встроенный отладчик адаптируется к используемым микросхемам, способен выставлять и работать с многофункциональными контрольными точками. Менеджер событий останавливает работу центрального процессора или вызывает иные действия, следуя определенным заранее условиям. Среда поддерживает режим трассировки, используемый для отладки программного кода, позволяет находить некоторые ошибки «невидимые» для других методов [2].

3. Прием и передача данных

Для беспроводного приема и передачи данных использовался последовательный синхронный интерфейс SPI для взаимодействия микроконтроллера MSP430F2274 с радиочастотным трансивером CC2500. Программная реализация последовательного синхронного интерфейса представляет собой библиотеку, предоставляемую фирмой TI [3]. Данная библиотека с кодом содержит базовые настройки регистров и функции для приема и передачи пакетов данных и является базовой платформой для

разработки приложений ориентированных на беспроводное соединение. В данном исследовании были изменены значения различных регистров в режиме SPI, отвечающих за фильтрацию пакетов принимаемых трансивером. Это было необходимо для корректного приема пакетов, отправленных широкопередаточным способом. Также для каждого узла сети задавался свой уникальный адрес, путем изменения значения.

При приеме данных трансивером CC2500 срабатывает прерывание на заданный порт микроконтроллера [4]. Это необходимо для того, чтобы предупредить микроконтроллер о приеме данных трансивером и затем через последовательный синхронный интерфейс передать их микроконтроллеру. Прерывание – подпрограмма, выполняемая при подаче сигнала на определенный порт микроконтроллера. После получения сигнала, если разрешено прерывание на данном выводе, основная программа прерывается на выполнения подпрограммы. По окончании выполнения подпрограммы, основная программа начинает работу с того места, где была прервана. Для получения необходимой информации о портах, на которые последует сигнал вызывающий прерывание в микроконтроллере, необходимо обратиться к схеме портов конкретного микроконтроллера. В рамках данного исследования использовалась спецификация, предоставляемая фирмой TI, к микроконтроллеру MSP430 [5].

4. Программная реализация протокола AODV

Протокол динамической маршрутизации AODV позволяет организовать беспроводную сенсорную сеть с ячеистой топологией. Данный протокол является реактивным протоколом, т. е. построение маршрута до нужного адресата происходит по требованию. В связи с этим в протоколе AODV минимизируется количество передаваемой служебной информации. Для выполнения поставленной задачи нет необходимости в полной реализации протокола AODV. Необходимым условием является реализация поиска неизвестного маршрута до узла назначения и динамическая работа с маршрутными таблицами.

Каждый узел содержит таблицу маршрутизации для передачи данных на нужный адрес. Формат таблицы маршрутизации приведен в таблице 1. В случае необходимости передачи пакетов данных на заданный адрес, устройство ищет запись с нужным адресом назначения в своей маршрутной таблице. При наличии такой записи происходит передача данных на узел, имеющий адрес «Адрес следующего узла». Каждая запись содержит поле «Время жизни маршрута». Это поле хранит время, по истечению которого запись будет удалена из таблицы маршрутизации. Каждый при отправке данных, текущее значение в поле «Время жизни маршрута» увеличивается на значение `ACTIVE_ROUTE_TIMEOUT`.

Помимо маршрутной таблицы, каждый узел ведет таблицу локальной истории. Ее формат представлен в таблице 2. Таблица локальной истории хранит записи обо всех недавно принятых RREQ запросах. Это необходимо для предотвращения повторной обработки RREQ пакетов. Запись в таблице локальной истории удаляется спустя PATH_DISCOVERY_TIME мс [6].

Таблица 1

Структура таблицы маршрутизации

Адрес получателя	Порядковый номер	Флаг корректности порядкового номера	Остальные AODV флаги	Количество скачков
MAC адрес	Значение (4 байта)	1 бит	3 бита	Значение (4 байта)

Таблица 2

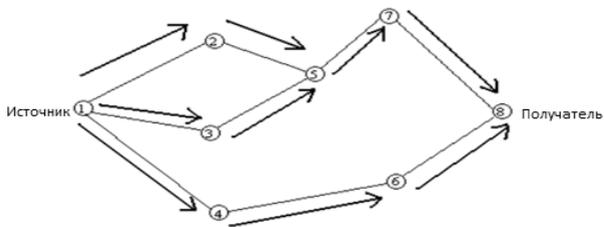
Структура таблицы локальной истории

Адрес отправителя	Идентификатор запроса
MAC адрес	Значение (4 байта)

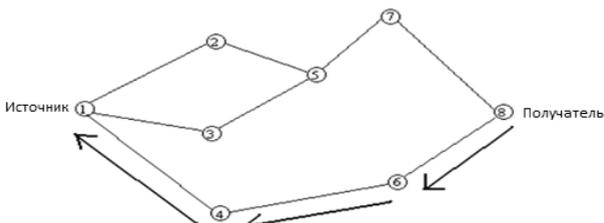
Таблицы маршрутизации и локальной истории должны быть представлены динамическими структурами данных для выделения и освобождения памяти в процессе работы устройства. Для программной реализации динамической структуры таблиц были разработаны библиотеки, содержащие структуры записей таблиц и функции для работы с таблицами, представляющие из себя двусвязный список, элемент которого соответствует одной записи в таблице.

В ситуации, когда узлу-отправителю неизвестен маршрут к необходимому ему адресату, выполняется алгоритм поиска маршрута до узла-назначения (рис. 2).

Для получения маршрута до адресата формируется пакет запроса маршрута Route Request (RREQ) и отправляется широковещательным способом всем соседним узлам сети. Формат пакета RREQ представлен в таблице 3. После отправки данного запроса, узел ожидает получения пакета ответа на запрос Route Reply (RREP), содержащий необходимый узлу-отправителю маршрут. Формат пакета RREP представлен в таблице 4.



Распространение пакета запроса маршрута RREQ



Передача пакета ответа на поиск маршрута RREP

Рис. 2. Поиск необходимого маршрута

Таблица 3

Формат пакета запроса маршрута (RREQ)

Адрес Отправителя	Идентификатор запроса	Адрес получателя	Порядковый номер отправителя	Порядковый номер получателя	Счетчик переходов
-------------------	-----------------------	------------------	------------------------------	-----------------------------	-------------------

Таблица 4

Пакет ответа на запрос поиска маршрута (RREP)

Адрес Отправителя	Адрес получателя	Порядковый номер получателя	Счетчик переходов	Время жизни пакета
-------------------	------------------	-----------------------------	-------------------	--------------------

Программная реализация данных пакетов представляет собой массив байт (тип `char` в языке Си), так как функции приема и передачи данных работают только с байтами. Значения полей «Порядковый номер получателя», «Порядковый номер отправителя», «Идентификатор запроса» и «Время жизни пакета» имеют целочисленный тип данных (`unsigned int` в Си), представленный в памяти 4 байтами. Для прямого и обратного преобразования типов используются функции, разработанные в ходе

исследования. При формировании пакетов RREQ и RREP используются функции библиотеки AODV.h. Данная библиотека разработана с целью упрощения формирования пакетов для нахождения маршрута.

При поступлении очередного пакета данных срабатывает прерывание на второй порт микроконтроллера для оповещения о принятии данных. Функция прерывания, вызываемая каждый раз при приеме нового пакета данных, реализует алгоритм обработки пакетов RREQ и RREP. Алгоритм обработки RREQ пакета представлен на рис. 3. При приеме трансивером пакета данных, срабатывает функция прерывания. В теле данной функции считываются данные из буфера обмена, после чего определяется тип пакета. В случае если был принят RREQ пакет, выполняется последовательность действий, описанная алгоритмом обработки RREQ запроса.

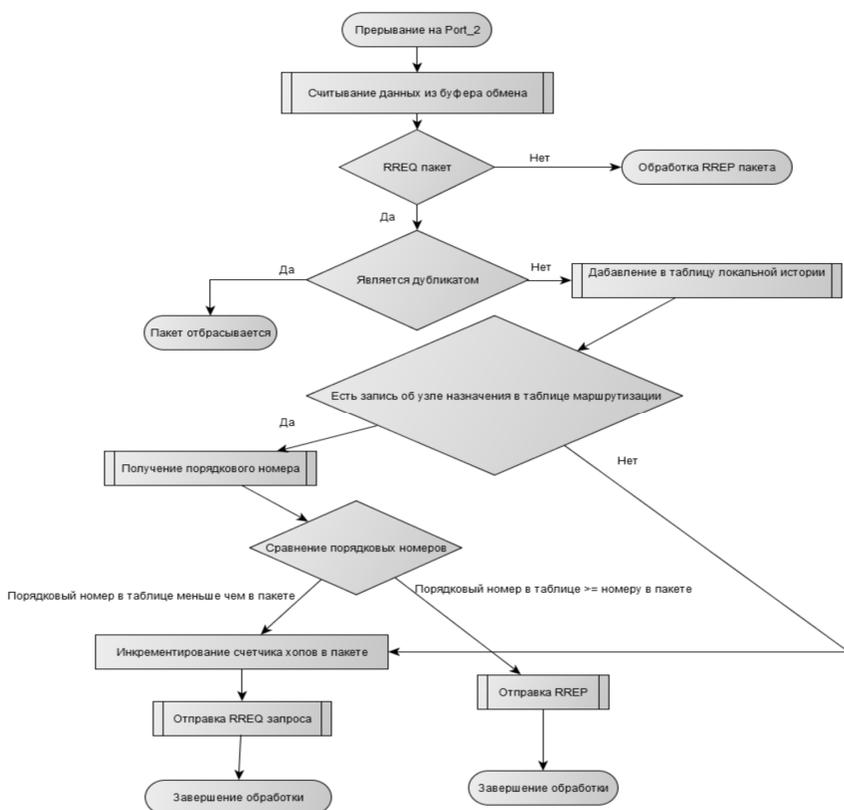


Рис. 3. Алгоритм обработки пакета RREQ

Пакет является дубликатом в случае, если в таблице локальной истории уже содержится запись с такими же значениями полей «Адрес отправителя», «Идентификатор запроса».

В случае если узел, обрабатывающий RREQ-запрос знает «свежий» маршрут к узлу назначения, отправляется пакет ответа на запрос поиска маршрута RREP. Маршрут «свежий», если «порядковый номер» в таблице маршрутизации больше или равен «порядковому номеру» в пакете запроса маршрута RREQ. В случае отсутствия «свежего» маршрута до узла назначения, инкрементируется счетчик переходов в пакете RREQ и отправляется дальше широковещательным способом [6].

Алгоритм обработки RREP пакета представлен на Рис. 4. Данный алгоритм описывает последовательность действий, в случае принятия RREP пакета.

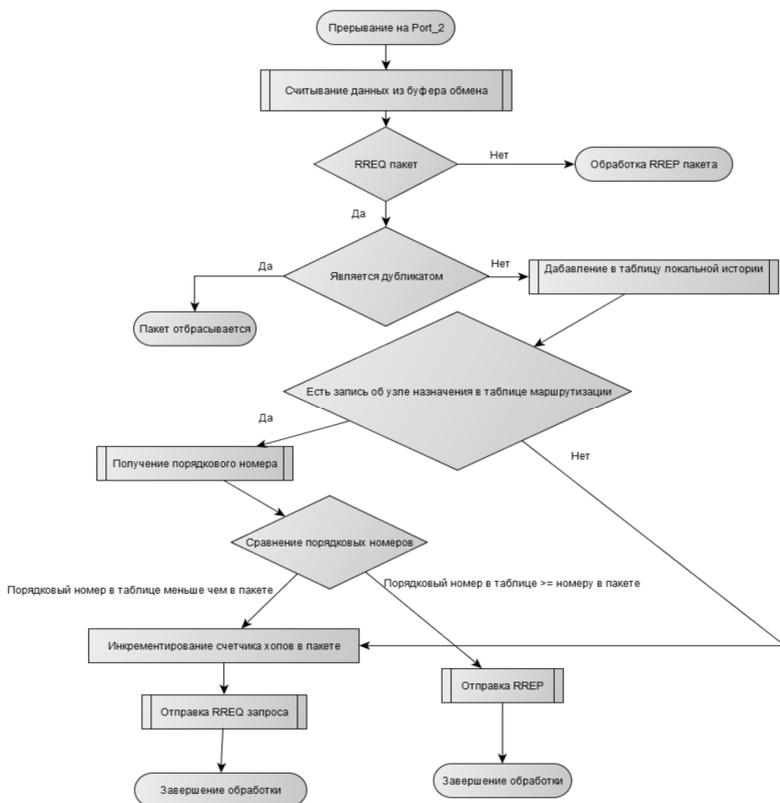


Рис. 4. Алгоритм обработки пакета RREP

Каждый узел принимающий пакет RREP делает запись в своей маршрутной таблице, в случае ее отсутствия, или обновляет значения полей «Порядковый номер», «Количество скачков». Если узел является отправителем RREQ запроса для поиска маршрута к узлу назначения, то отправитель получает необходимый маршрут из пакета RREP и отправляет данные. В противном случае, узел, получивший пакет RREP, находит запись в своей маршрутной таблице до узла назначения и отправляет пакет на адрес соответствующий адресу в поле «Адрес следующего узла».

Заключение

Данная статья посвящена проблеме построение сенсорных сетей на базе микроконтроллеров MSP430. Приведена необходимая информация для разработки программного обеспечения, на основе которого разработчик может развертывать беспроводные mesh-сети. Описаны основные особенности реализации подобного ПО, а также средство разработки для программирования микроконтроллера. Основной задачей данного исследования являлась программная разработка алгоритмов поиска неизвестного маршрута узлами сети. Для этого были выбраны и реализованы алгоритмы протокола AODV, а также разработка библиотек, содержащих функции и структуры пакетов запросов, для корректной и удобной работы алгоритмов построения маршрута от узла-отправителя до узла-назначения. Представленные выше алгоритмы программно реализованы и протестированы. Результаты тестов показали полную работоспособность, разработанных в ходе исследования алгоритмов, функций и структур данных.

Список литературы

1. Обзор архитектуры MSP430 [Электронный ресурс]. – Режим доступа : <http://www.gaw.ru/html.cgi/txt/app/micros/msp430/slaa217.htm>
2. Среда проектирования Code Composer Studio [Электронный ресурс]. – Режим доступа : http://cxem.net/software/code_composer_studio.php
3. Последовательный синхронный интерфейс SPI [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/Serial_Peripheral_Interface
4. Обзор радиочастотного трансивера [Электронный ресурс]. – Режим доступа : http://andromega.narod.ru/doc/cc2500_ds_1_2_rus_part.pdf
5. Евстифеева А. В. Семейство микроконтроллеров MSP430x2xx. Архитектура, программирование, разработка приложений / А.В. Евстифеева. – М. : Додэка XXI, 2010. – 544 с.
6. Спецификация протокола AODV [Электронный ресурс]. – Режим доступа : <https://www.ietf.org/rfc/rfc3561.txt>

ПОСТРОЕНИЕ РОБОТА НА БАЗЕ МИКРОКОНТРОЛЛЕРА RASPBERRY PI

М.С. Мишанин

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Микроконтроллеры используются в настоящее время промышленно во множестве устройств. При этом, несмотря на относительную дешевизну, в индивидуальном порядке (на любительском уровне, для прототипирования различных устройств и т. п.) до сих пор микроконтроллеры используются довольно редко. Это объясняется высоким порогом вхождения в этой области (как минимум надо уметь программировать, разбираться в схемотехнике, возможно, «дружить» с паяльником и т. д.). Для упрощения применения микроконтроллерной техники в последние годы появились готовые платы, которые помимо, собственно, микроконтроллера, включают его обвязку, для того, чтобы упростить сопряжение микроконтроллера с датчиками, управляемыми устройствами, компьютером и т.д. Наиболее известной платой такого рода является Arduino, построенная на микроконтроллерах AVR. Несмотря на все достоинства Arduino (широкое распространение, дешевизна, множество устройств сопряжения и т. д.), данная плата может использоваться только в относительно простых устройствах, т.к. возможности микроконтроллеров AVR весьма ограничены (достаточно сказать, что размер ОЗУ Arduino составляет 2-8 Кб). Для применений, требующих значительных вычислительных ресурсов, например, анализ видео-потока и т. п., необходимо использовать другие решения.

Для таких приложений также существует несколько вариантов готовых плат, но наибольшее распространение получила плата Raspberry Pi на базе ARM-процессоров. По сути плата Raspberry Pi является портативным компьютером (размер банковской карты) с дополнительными интерфейсами ввода-вывода общего назначения для подключения различных устройств. Соответственно, на эту плату можно установить полноценную операционную систему и использовать любые удобные средства разработки.

Мотивацией данной работы являлось изучение возможностей Raspberry Pi при применении в качестве встраиваемой платы в устройствах, требующих значительные вычислительные ресурсы. В

качестве примера такого устройства в данной работе рассматривается робот, который должен собирать кубик Рубика. Для того, чтобы задействовать возможности Raspberry Pi, предполагается, что:

- устройство должно быть автономным (для вычислений не использовать другие ЭВМ);
- для получения начального состояния кубика Рубика должна использоваться камера, подключенная к Raspberry Pi (где происходит обработка «картинки» с камеры);
- управляемые устройства (серво-привод, шаговый двигатель) также подключены непосредственно к плате Raspberry Pi.

1. Конструкция робота

В данной работе использовалась плата Raspberry Pi 2B. Данная модель обладает четырехъядерным процессором ARM Cortex-A7 с частотой 900 МГц, ОЗУ 1 ГБ, GPIO в количестве 40 пинов. Также эта платформа оснащена четырьмя USB портами, одним HDMI портом и одним Ethernet портом [1].

Механическая конструкция робота во многом повторяет существующие аналоги (см. рис. 1) и состоит из:

- наклонной вращающейся платформы-«корзинки», на которой размещается кубик Рубика с фиксацией нижнего слоя
- манипулятора-«руки», который может как фиксировать верхние два слоя кубика, так и толкать кубик, тем самым переворачивая его на бок в «корзинке».

Вращение «корзинки» происходит с помощью шагового двигателя EM-519. Для подключения шагового двигателя использовался драйвер двигателей L293D [2], который позволяет подавать напряжение на концы каждой обмотки двигателя, а также подключать стороннее питание для него, так как питание подаваемого микроконтроллером недостаточно.

«Рука» приводится в движение сервоприводом MG995. Для подключения сервопривода использовалось питание, предоставляемое платой Raspberry Pi. Управление сервоприводом осуществляется с помощью широтно-импульсной модуляции (ШИМ) [2]. ШИМ – это операция получения изменяющегося аналогового значения посредством цифровых прямоугольных импульсов.

Веб-камера подключена в USB-разъем Raspberry Pi (предварительно необходимо в настройках платы включить использование камеры) [2].

Схема подключения устройства показана на рис. 2.

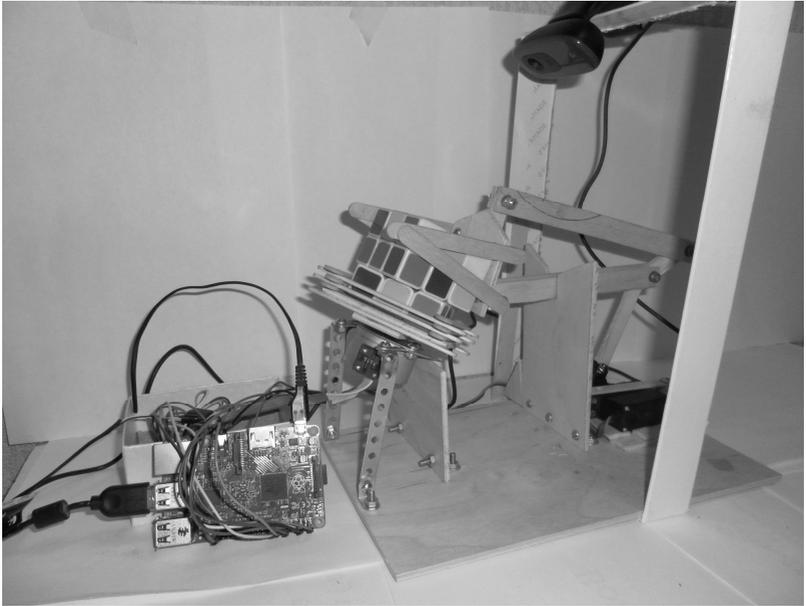


Рис. 1. Собранный робот

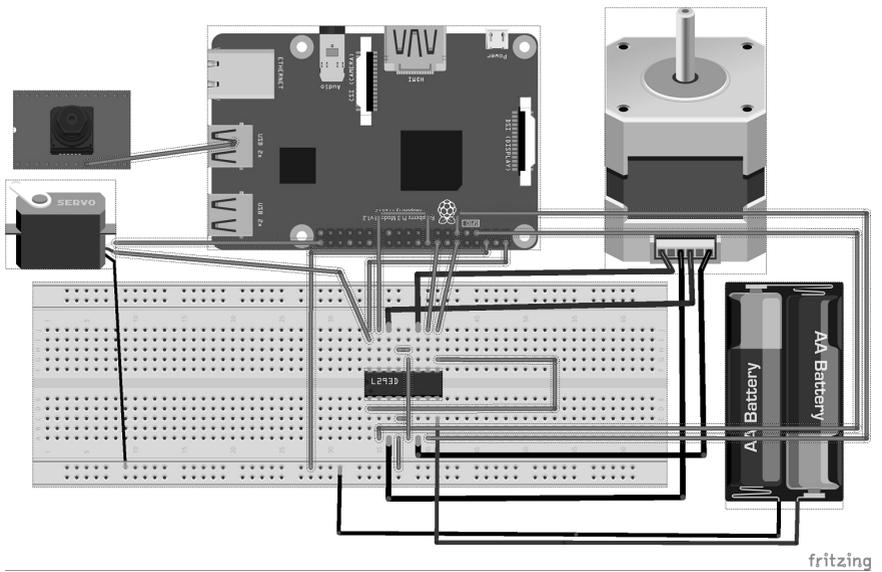


Рис. 2. Схема подключения устройств к Raspberry Pi

2. Программная реализация

Языком программирования выбран Python, т. к. данный язык является высокоуровневым и удобным, предустановлен в стандартной прошивке Raspberry Pi, а также обладает необходимым набором библиотек для разработки робота.

Робот выполняет следующие элементарные операции:

– «Повернуть кубик в вертикальной плоскости»: подается сигнал сервоприводу для поворота кубика в вертикальной плоскости, осуществляемый толчком кубика «рукой» («Толкнуть кубик рукой»);

– «Повернуть кубик в горизонтальной плоскости»: подается сигнал шаговому двигателю для поворота кубика в нужную сторону в горизонтальной плоскости («Повернуть корзинку в нужную сторону»);

– «Повернуть слой» – микроконтроллер подает сигнал сервоприводу для фиксации кубика «рукой» («Зафиксировать рукой кубик»), после чего подается сигнал для поворота двигателя в нужную сторону («Повернуть корзинку в нужную сторону»).

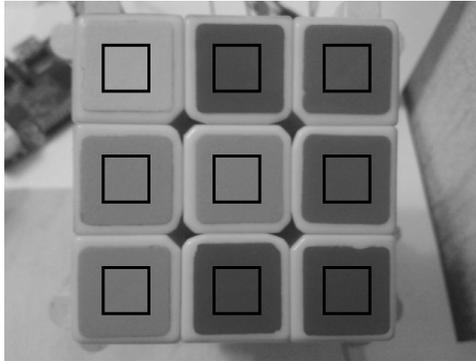
Задача распознавания начального состояния кубика включает в себя несколько подзадач:

- получение изображений каждой стороны;
- считывание цветов всех элементов кубика;
- кластеризация по цветам;
- представление начального состояния в UFRDBL-нотации, которая используется в алгоритме сборки (описана ниже).

Для получения изображений каждой стороны кубика с помощью библиотеки OpenCV выполняется подключение к видеопотоку камеры и считывание текущего изображения. Пример такого изображения приведен на рис. 3.

На данном рисунке также отмечены области, которые используются для считывания цвета каждого элемента стороны кубика. Цвет элемента считается усреднением цвета пикселей в заданной области. Так сделано, чтобы сгладить возможные артефакты на изображении (блики и т.п.). Так как камера закреплена относительно положения кубика жестко, местоположение областей, используемых для определения цвета, заданы в виде константных значений. Размер областей выбран меньше размеров элементов кубика на получаемом изображении для того, чтобы небольшие отклонения положения кубика или камеры из-за нежесткости конструкции не влияли на правильность позиционирования данных областей относительно элементов кубика на изображении.

Промежуточным результатом считывания всех сторон кубика (для этого в процессе считывания его необходимо несколько раз перевернуть), является массив из 54 цветов элементов кубика.



Прямоугольниками отмечены области, используемые для вычисления цветов отдельных элементов каждой стороны кубика

Рис. 3. Изображение кубика Рубика, получаемое с камеры

Для кластеризации этих пикселей на группы используется алгоритм *k* ближайших соседей [3]. Каждый цвет рассматривается как точка в трехмерном пространстве с координатами, соответствующими RGB-компонентам цвета. Группировка происходит относительно центральных элементов кубика, т.е. считается условное расстояние между каждым центром и всеми элементами кубика, после чего выбираются 8 ближайших, не считая сам центральный. После кластеризации формируется массив, состоящий из 54 символов, указывающих на принадлежность элемента к какой-либо стороне в собранном состоянии.

Последним этапом является перевод данного массива в строку, соответствующую UFRDBL-нотации.

Выполнение поиска последовательности операций для сборки кубика Рубика является алгоритмически сложной задачей, поэтому реализация алгоритма сборки была выполнена на языке C++ в виде Python-модуля.

Реализованный алгоритм использует UFRDBL-нотацию представления состояния кубика. Суть данной нотации заключается в описании цветов элементов кубика в терминах цветов центральных элементов каждой стороны кубика, которые обозначаются следующими буквами:

- F (Front) – передняя грань;
- U (Up) – верхняя грань;
- R (Right) – правая грань;
- D (Down) – нижняя грань;
- L (Left) – левая грань;
- B (Back) – задняя грань.

На рис. 4 представлена развертка кубика с указанием граней.

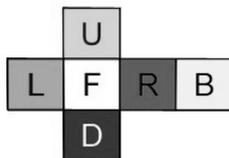


Рис . 4. Развертка кубика

В UFRDBL-нотации принято описывать цвета элементов не по сторонам кубика Рубика, а по отдельным физическим элементам, из которых состоит кубик. Сначала описываются центральные элементы на каждом ребре (состоят из 2-х цветов, 12 элементов), затем угловые элементы (состоят из 3-х цветов, 8 элементов). Центральные элементы не описываются, т.к. они никогда не меняют своего цвета, соответственно, любое описание состояния кубика Рубика состоит из 48 символов. В UFRDBL-нотации принят следующий порядок описания элементов (отдельные элементы отделены друг от друга пробелами):

UF UR UB UL DF DR DB DL FR FL BR BL (центральные элементы)

UFR URB UBL ULF DRF DFL DLB DBR (угловые элементы).

Соответственно, собранный кубик Рубика имеет такое же описание состояния.

Повороты граней кубика Рубика также определяются буквами UFRDBL: каждая буква определяет поворот на 90° по часовой стрелке соответствующей грани кубика. К буквам можно приписывать цифры (обозначает кол-во поворотов на 90°) или символы «+» и «-».

Изменение состояния кубика в UFRDBL-нотации можно описать в виде порядка перестановки соответствующих символов в строке описания состояния кубика. Этот порядок можно как вычислить, так и задать константно.

Реализованный алгоритм работает по шагам. На каждом шаге состояние кубика Рубика приводится к требуемому на данном шаге состоянию. Целевые состояния в UFRDBL-нотации заданы следующим образом:

- 1) ??????????????????FL?????????????ULF?????????????
- 2) UF????UL?????????????FL?????????????ULF?????????????
- 3) UFUR??UL?????????FRFL????UFR?????ULF?????????????
- 4) UFUR??ULDF?????FRFL????UFR?????ULFDRFDFL???????
- 5) UFURUBULDF?????FRFL????UFRURB??ULFDRFDFL???????
- 6) UFURUBULDF?????FRFL????UFRURBUBLULFDRFDFL???????
- 7) UFURUBULDF?????FRFLBR??UFRURBUBLULFDRFDFL???????
- 8) UFURUBULDF?????FRFLBRBLUFRURBUBLULFDRFDFL???????

- 9) UFURUBULDF?????FRFLBRBLUFRURBUBLULFDRDFDLDLB???
- 10) UFURUBULDF?????FRFLBRBLUFRURBUBLULFDRDFDLDLBDBR
- 11) UFURUBULDFDR????FRFLBRBLUFRURBUBLULFDRDFDLDLBDBR
- 12) UFURUBULDFDRDB??FRFLBRBLUFRURBUBLULFDRDFDLDLBDBR
- 13) UFURUBULDFDRDBDLFRFLBRBLUFRURBUBLULFDRDFDLDLBDBR

В приведенном списке вопросами обозначены элементы, цвет которых на очередном шаге не имеет значения. На каждом следующем шаге правильно должно располагаться большее кол-во элементов кубика Рубика, т. е. его сборка выполняется последовательно. Поиск последовательности шагов по переводу кубика Рубика к требуемому состоянию на очередном шаге осуществляется рекурсивным перебором возможных вращений кубика с ограничением на глубину рекурсии. При этом используются только D-, B-, U-, F-вращения.

Кроме этого для ускорения поиска решения после каждой операции происходит сравнение состояния кубика Рубика с набором возможных состояний, сборка из которых известна. Набор этих состояний строится в виде рекурсивной последовательности операций над «собранным» состоянием кубика с ограничением на глубину рекурсии. Таких состояний при глубине рекурсии в 10 шагов получается 171136. Если в процессе рекурсивного поиска на любом шаге получилось одно из этих состояний, то оставшаяся часть сборки кубика Рубика уже известна – решение найдено.

Полученное решение остается корректно перевести в элементарные операции, поддерживаемые роботом, и выполнить.

Заключение

Raspberry Pi – платформа, которая может использоваться во встраиваемых автоматизированных системах со значительными требованиями к вычислительным ресурсам.

В результате проделанной работы были изучены различные способы работы с данной платой и варианты подключения внешнего оборудования. В качестве примера был разработан прототип робота, собирающий кубик Рубика и отвечающий поставленным требованиям.

Список литературы

1. Grimmett R. Raspberry Pi Robotic Projects / R. Grimmett. – Birmingham : Packt Publishing, 2014. – 278 с.
2. Петин В. Микрокомпьютеры Raspberry Pi. Практическое руководство / В. Петин. – СПб : БХВ-Петербург, 2015. – 240 с.
3. Алгазинов Э. К. Анализ и компьютерное моделирование информационных процессов и систем / Э. К. Алгазинов, А. А. Сирота. – М. : Диалог-МИФИ, 2009. – 416 с.

ИЗУЧЕНИЕ РЕЧЕВЫХ ТЕХНОЛОГИЙ И ПОСЛЕДУЮЩЕЕ ИХ ПРИМЕНЕНИЕ В СФЕРЕ ЗАПИСИ РЕЧЕВОЙ ИНФОРМАЦИИ

П.А. Морозов

Студент

С.А. Зуев

Ассистент

Введение

В современном мире человек постоянно сталкивается с задачей преобразования аналоговых сигналов в цифровую форму. Одним из примеров такого преобразования могут служить технологии распознавания речи, которое все чаще применяется в сферах науки, бизнеса и даже творчества. Это вносит неоспоримое удобство при общении между пользователем и компьютером, ведь оно становится более естественным, так как взаимодействие может быть осуществлено не только посредством тонового набора, но и с помощью голосовых команд. Можно с уверенностью сказать, что уже в скором времени голосовые интерфейсы практически не будут отличаться по надежности от классических способов ввода различных видов информации в память компьютера.

Распознавание речи — процесс преобразования речевого сигнала в текстовую информацию. Системы распознавания речи классифицируются по размеру используемого словаря (ограниченный набор слов или словарь большого размера), по зависимости от диктора (дикторозависимые и дикторонезависимые системы), по типу речи (слитная и раздельная речь), по назначению (системы диктовки, командные системы).

1. Цель работы

Цель данной статьи — ознакомление и анализ существующих технологий распознавания речи, их преимущества и недостатки, выбор оптимального, с практической точки зрения решения, подробное изучение алгоритма распознавания речи для выбранной технологии и описание способа реализации веб-приложения с поддержкой функции распознавания речи.

Приложение, в свою очередь, представляет собой текстовый редактор, обладающий возможностью, как стандартного ввода текста, так и голосового. Для расширения способов ввода информации, в приложение встроен графический редактор. Чтобы облегчить поиск необходимых пользователю документов, приложение имеет базу данных.

2. Анализ

Реализация собственной системы распознавания речи с открытым исходным кодом является трудоемкой и затратной задачей, с точки зрения необходимых ресурсов.

Существуют различные решения при применении подхода API (API — Application Programming Interface) — библиотеки распознавания речи с закрытым исходным кодом, предоставляющие разработчикам доступ к использованию технологий при работе с данными:

- Yandex Speech Kit [1];
- Google Speech Recognition API [2];
- Microsoft Speech Platform [3];
- Nuance Dragon Mobile SDK [4].

Yandex Speech Kit рекомендуется использовать, если приложение рассчитано на российскую аудиторию, а русский язык выступает в роли первичного языка использования.

Google Speech Recognition API рекомендуется использовать, если приложение рассчитано на международную аудиторию, и есть необходимость в распознавании большого количества языков.

Microsoft Speech Platform рекомендуется использовать только при разработке ПО на базе платформы Windows.

Nuance Dragon Mobile SDK хоть и является лидером в области языковых и голосовых решений, но все же библиотеку рекомендуется использовать при создании масштабного коммерческого приложения, так как Nuance обладает ограниченным бесплатным функционалом.

Для реализации данной работы была выбрана библиотека Yandex Speech Kit, так как тестирование показало, что процент правильности распознавания русскоязычной речи у этой библиотеки выше, чем у остальных.

3. Изучение алгоритма распознавания речи Yandex Speech Kit

Yandex Speech Kit — публичный API для распознавания речи. Данная библиотека доступна разработчикам под iOS, Android, Windows Phone и для других платформ с использованием веб-доступа. Опишем детали работы технологии распознавания русскоязычной речи Яндекса.

Акустическая модель принимает на вход звук, а на выходе выдает распределение вероятностей, по так называемым, сенамам. Звук нарезается участками по 30 мс («фреймами»). Шаг нарезки составляет 10 мс, поэтому соседние фреймы часто пересекаются. Полученные величины подвергаются дискретному косинусному преобразованию (DCT), и в результате получается вектор, состоящий из 13 вещественных чисел (MFCC – Mel Frequency Cepstral Coefficients). Они могут использоваться

как вход акустической модели, но чаще всего подвергаются множеству дополнительных преобразований.

Для улучшения распознавания речи требуется тренировка акустической модели — сложный и многоступенчатый процесс. Для тренировки используются алгоритм семейства Expectation-Maximization — алгоритм Баума-Велша (необходим для оценки параметров скрытых марковских моделей). Expectation заключается в использовании имеющейся модели для вычисления максимума математического ожидания функции правдоподобия, а Maximization используется для изменения параметров модели таким образом, чтобы максимизировать эту оценку. На ранних этапах тренировки используются простые акустические модели: на вход подаются MFCC-Features, фонемы рассматриваются вне контекстной зависимости, а для моделирования вероятности эмиссии в скрытой марковской модели (HMM — Hidden Markov Model) используется смесь гауссиан с диагональными матрицами ковариаций (Diagonal GMMs — Gaussian Mixture Models). Результаты каждой предыдущей акустической модели являются стартовой точкой для тренировки более сложной модели, с более сложным входным сигналом.

На данный момент существует большое количество способов улучшения акустической модели, однако наиболее эффективным является переход от GMM-модели к DNN (Deep Neural Network). Такой переход повышает качество распознавания речи практически в два раза. Нейронные сети лишены многих ограничений, характерных для GMMs, и обладают лучшей обобщающей способностью. Кроме того, акустические модели на нейронных сетях более устойчивы к шуму и имеют лучшее быстроедействие.

Нейронная сеть для акустического моделирования тренируется в несколько этапов. Для инициализации нейросети используется стек из ограниченных машин Больцмана (RBM — Restricted Boltzmann Machines). RBM это стохастическая нейросеть, которая способна тренироваться без учителя. Выученные ей веса детально отражают структуру речи. По сути, RBM — механизм извлечения признаков (Feature Extractor) — полученная генеративная модель оказывается отличной стартовой точкой для построения дискриминативной модели. Дискриминативная модель, в свою очередь, тренируется с использованием классического алгоритма обратного распределения ошибки, при этом к ней применяется ряд технических приемов, которые улучшают сходимость и предотвращают переобучение (Overfitting). В итоге, на входе нейросети мы имеем несколько MFCC-Features (центральный фрейм подлежит классификации, остальные образуют контекст), на выходе получаем около 4000 нейронов, соответствующих различным сенам — элементарным единицам языка.

Это и есть фонетический словарь, с которым работает технология распознавания речи Яндексa.

Теперь рассмотрим процесс декодирования. Для решения задачи распознавания спонтанной речи с большим словарем, требуется структура, позволяющая объединить все возможные предложения, которые может распознать система (WFST — Weighted Finite-State Transducer). На вход этого автомата подаются сеноны, на выходе — слова. Процесс декодирования сводится к тому, чтобы выбрать лучший путь в этом автомате, и предоставить выходную последовательность слов, которая будет соответствовать этому пути. Результат процесса декодирования — список гипотез, который может быть подвергнут дальнейшей обработке. Результирующий список, который вернется пользователю, будет отсортирован по значению confidence — степени нашей уверенности в том, что распознавание прошло правильно. Нередко остается всего одна гипотеза, в этом случае приложение-клиент сразу переходит к выполнению голосовой команды.

4. Подключение пользовательского интерфейса

Для удобства использования приложения необходимо подключить текстовый редактор, в который будет встроена функция распознавания речи. При этом появляется возможность оперативного редактирования ошибочно распознанных слов, а наличие автоматической корректировки с встроенным словарем, позволяет делать это и в автоматическом режиме в большинстве практических случаев. Для расширения возможности ввода информации в текстовый редактор внедрен графический редактор. А для хранения документов возникает необходимость подключить встроенную в приложение базу данных.

Рассмотрим примененные нами при решении этой задачи технологии:

- TinyMCE Advanced [5] — плагин для внедрения в приложение визуального редактора HTML на основе JavaScript.

- Canvas Painter — программное решение, позволяющее подключить к приложению графический редактор. При чем, графический редактор имеет открытый исходный код.

- В приложении реализована база данных для хранения пользовательских документов. Управление базой данных происходит с помощью СУБД MySQL. А для динамического отображения данных в базе используется React JS [6] — JavaScript-библиотека для создания пользовательских интерфейсов, разработанная компанией «Facebook».

5. Структура реализации

Для реализации программы использовался Play [7] — современный фреймворк, ориентированный на создание веб-приложений. Play написан

на языке программирования Scala, в качестве языка разработки поддерживает как Scala, так и Java. Play работает в асинхронном стиле с помощью библиотеки Akka, использует набор шаблонов программирования Model-View-Controller в архитектуре создаваемых приложений. В Play встроены фреймворки Ebean для работы с базами данных, и JUnit для тестирования. Шаблоны страничек пишутся на Scala. Приложение запускается на встроенном Netty-сервере. В данном проекте используется язык разработки Java 8.

Для более наглядного представления работы приложения построим диаграмму размещения (рис. 1).

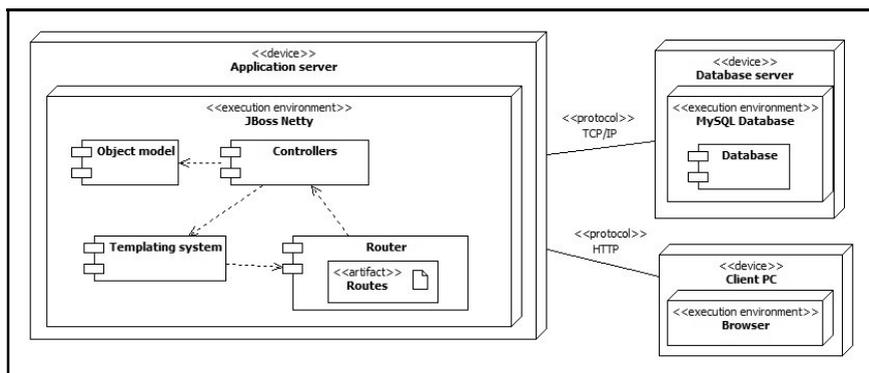


Рис. 1. Диаграмма размещения

Диаграмма размещения (Deployment diagram) отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе.

На диаграмме присутствует сервер, клиент и БД. Сервер содержит в себе контроллер, использующийся для связи базы данных с графическим представлением, роутер, связывающий адрес в браузере с контроллером, и модель, отображающую сущности БД. Клиент, с помощью HTTP-запроса, обращается к нашему Netty-серверу. Запросы посылает браузер. Сервер, в процессе обработки, берет информацию из базы данных с помощью ORM-библиотеки Ebean.

Приложение имеет клиентско-серверную структуру. Опишем реализацию серверной и клиентской части проекта.

Серверная часть (Backend), в приложении представляет собой совокупность взаимодействия моделей, контроллера и шаблонов, что

позволяет связать базу данных с графическим представлением. Построим ER-диаграмму БД (рис. 2).

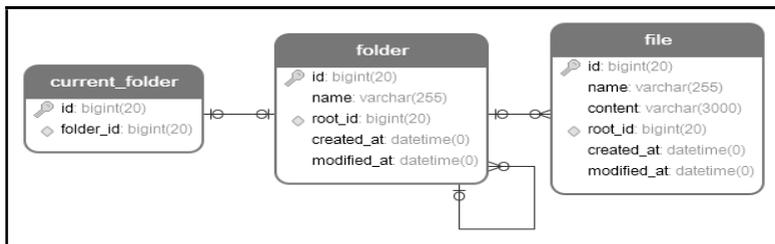


Рис. 2. ER-диаграмма

- «Current folder» — хранит информацию о текущей папке для файлового менеджера.
- «Folder» — хранит информацию с именем папки, ссылками на корень папки и на файлы внутри папки.
- «File» — хранит информацию о документах, и ссылки на папки в которых они хранятся.

Клиентская часть (Frontend) в приложении представляет собой пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. К клиентской части приложения относится подключение Yandex Speech Kit, Canvas Painter, TinyMCE Advanced. К клиентской части библиотеки относится подключение React JS.

В «Index.Scala.html» происходит подключение библиотеки Yandex Speech Kit (рис. 3).

```
<script type="text/javascript" src="//download.yandex.ru/webspeechkit/webspeechkit-1.0.0.js"></script>
<script type="text/javascript" src="//download.yandex.ru/webspeechkit/webspeechkit-settings.js"></script>
<script src="@routes.Assets.versioned("javascripts/index.js")" type="text/javascript"></script>
```

Рис. 3. Подключение библиотеки Yandex

Далее, происходит инициализация библиотеки Yandex Speech Kit (рис. 4).

```
var speech = ya.speechkit.SpeechRecognition();

function insertYourContent(text) {
    var output = '';
    if (text.indexOf('<img') == -1) {
        if (prevWord.trim() == '.' || prevWord.trim() == '?' || prevWord.trim() == '!') {
            console.log('cap!');
            output = ' ' + text.capitalize();
        } else if (prevWord == '') {
            output = text.capitalize();
        } else {
            console.log(prevWord);
            if (names.filter(function(val, i){
                //console.log(text);
                if (text.trim().toLowerCase() == val.toLowerCase().trim()) {
                    return true;
                } else {
                    return false;
                }
            }).length > 0) {
                output = ' ' + text.capitalize();
            } else {
                output = ' ' + text;
            }
        }
        prevWord = text;
    } else {
        output = text;
    }
    tinyMCE.activeEditor.execCommand('mceInsertContent', false, output);
    checkSigns();
}
```

Рис. 4. Инициализация библиотеки Yandex

В классе распознавания передаются параметры приложения Speech Kit (API key), язык распознавания, модель (freeform). Также, передаются и callback-функции, куда поступает вся информация от Yandex Speech Kit. «DataCallback» принимает данные о результатах распознавания в массиве «Words», после этого результат добавляется в текстовый редактор через функцию «InsertYourContent». Класс «ya.speechkit.SpeechRecognition» работает с потоком аудио в формате PCM16.

6. Тестирование функциональности

После нажатия на кнопку, начинается процесс записи сигнала. Для тестирования, произнесем определение термина «модуляция» (рис. 5).

После нажатия на кнопку графического редактора открывается окно рисования. Для примера, зарисуем формулу несущего модулируемого сигнала (рис. 6).

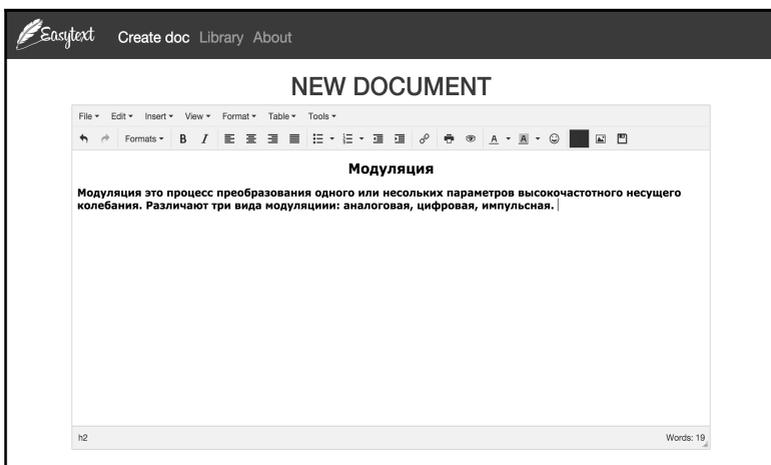


Рис. 5. Процесс распознавания речи



Рис. 6. Окно рисования

После окончания редактирования документа, при нажатии на кнопку дискеты, откроется окно сохранения файла (рис. 7).

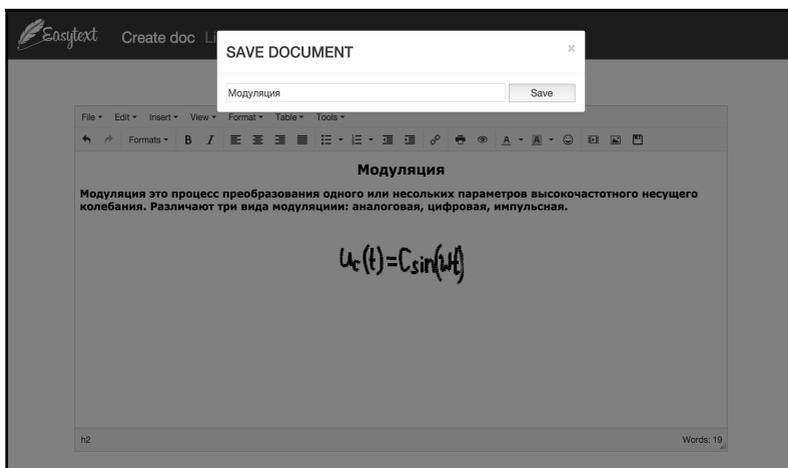


Рис. 7. Окно сохранения файла

После сохранения файла перейдем в раздел библиотеки. Там и будут храниться все созданные документы (рис. 8).

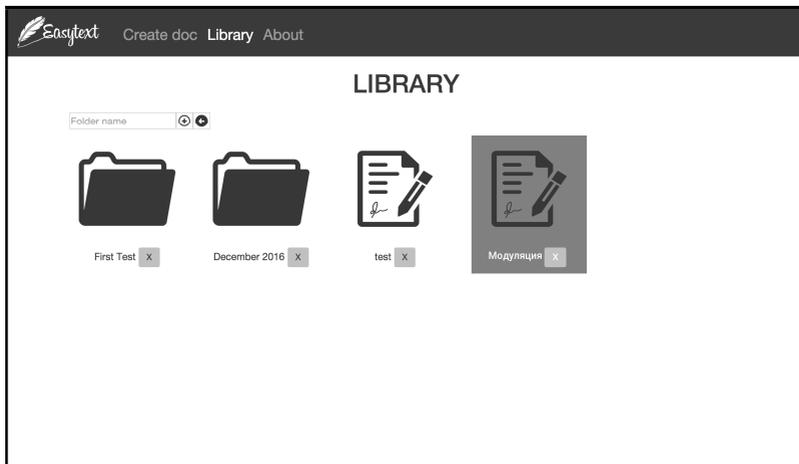


Рис. 8. Главная страница библиотеки

Заключение

Данная статья посвящена анализу существующих речевых библиотек Speech Kit, а также разработке и внедрению одной из библиотек в веб-приложение. Описан алгоритм работы библиотеки Yandex Speech Kit.

Итогом данной работы является разработка и построение полностью рабочего веб-приложения, представляющее собой текстовый редактор с возможностью как стандартного ввода текста, так и голосового. Способы ввода данных расширены посредством внедрения в приложение графического редактора. Подключена база данных для хранения документов. Использование получившегося веб-приложения позволяет решить проблему скорости перевода большого объема текстовой информации в электронный вид, что при условии постоянно растущего документооборота является несомненным преимуществом.

Список литературы

1. Комплекс речевых технологий Яндекса [Электронный ресурс] : инф. разработ. — Режим доступа : <https://tech.yandex.ru/speechkit/>
2. Cloud Speech Google API [Электронный ресурс] : инф. разработ. — Режим доступа : <https://cloud.google.com/speech/>
3. Microsoft Speech Platform [Электронный ресурс] : инф. разработ. — Режим доступа : <https://msdn.microsoft.com/office/>
4. Enables the advanced features TinyMCE [Электронный ресурс] : инф. разработ. — Режим доступа : <https://ru.wordpress.org/plugins/tinymce-advanced/>
5. React [Электронный ресурс] : инф. разработ. — Режим доступа : <https://facebook.github.io/react/>
6. Nuance Developers [Электронный ресурс] : инф. разработ. — Режим доступа : <https://developer.nuance.com/>
7. Web Framework For Java and Scala [Электронный ресурс] : инф. разработ. — Режим доступа : <https://www.playframework.com/>

СИСТЕМЫ БИЛИНЕЙНЫХ УРАВНЕНИЙ

Е.Ю. Морозов

Студент

А.В. Лобода

Профессор

Введение

Для билинейных систем не существует теории, которая бы позволила решать их в общем виде [1]. Тем не менее, в некоторых математических задачах, к примеру, в задаче описания аффинно-однородных поверхностей в пространстве S^3 , возникает такая потребность. В однородных билинейных системах, которые появляются в указанной задаче, особый

интерес представляет наличие ненулевых решений. В данной работе рассматриваются частные случаи билинейных систем и выводится критерий, позволяющий определить, имеет ли система ненулевые решения.

1. Постановка задачи

Система билинейных уравнений (или билинейная система) – это система следующего вида:

$$\sum_{j=1}^m \sum_{k=1}^n a_{jk}^i r_j s_k = d_i, i = \overline{1, l}, \quad (1)$$

где m, n – число неизвестных в векторах \mathbf{r} и \mathbf{s} , l – число уравнений и a_{jk}^i – действительные числовые коэффициенты (верхний индекс показывает номер уравнения) [1]. Для такой системы введём компактную запись её параметров $(l; m, n)$.

Билинейную систему можно записать в альтернативном виде:

$$\mathbf{r}^T \cdot A_i \cdot \mathbf{s} = d_i, \quad (2)$$

где $\mathbf{r} \in R^m$, $\mathbf{s} \in R^n$ – векторы неизвестных, $A_i \in R^{m \times n}$ – матрица билинейной формы i -го уравнения.

Для каждой билинейной системы можно составить *обобщенную матрицу системы*, составленную из развернутых в строки матриц билинейных форм каждого уравнения:

$$A = \begin{pmatrix} a_{11}^1 & \cdots & a_{m1}^1 & \cdots & a_{mn}^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{11}^l & \cdots & a_{m1}^l & \cdots & a_{mn}^l \end{pmatrix} \in R^{l \times mn}.$$

Ранг системы билинейных форм (1) совпадает с рангом этой матрицы. В этой работе будут рассматриваться только системы полного ранга.

Билинейная система называется *однородной*, если правая часть равна нулю, т. е. $d_i = 0$. У таких систем всегда есть тривиальные решения – решения, при которых один из наборов в переменных является нулевым, т. е. $\mathbf{r} = \mathbf{0}$ или $\mathbf{s} = \mathbf{0}$.

В данной работе будет рассматриваться вопрос о наличии именно нетривиальных решений в однородных системах билинейных уравнений. Как правило, системы с большим количеством уравнений имеют только тривиальные решения, поэтому системы, имеющие нетривиальные решения будем называть *нерегулярными*.

В уже упомянутой задаче описания аффинно-однородных поверхностей в пространстве S^3 возникает потребность в установлении

факта регулярности системы. Один из возникающих сложных случаев связан с

$$m = n, l = 2m - 1.$$

Особый интерес в уже упомянутой задаче представляют именно нерегулярные системы и в основном случае (15;8,8) нетривиальные решения пока не описаны. В связи с этим, в данной работе изучаются более простые случаи. Самый простой из них – (1;1,1), является тривиальным, и рассматривать его отдельно не имеет смысла.

2. Случай (3;2,2)-систем

В следующем случае возникает система вида:

$$\begin{cases} a_{11}r_1s_1 + a_{12}r_1s_2 + a_{13}r_2s_1 + a_{14}r_2s_2 = 0, \\ a_{21}r_1s_1 + a_{22}r_1s_2 + a_{23}r_2s_1 + a_{24}r_2s_2 = 0, \\ a_{31}r_1s_1 + a_{32}r_1s_2 + a_{33}r_2s_1 + a_{34}r_2s_2 = 0. \end{cases} \quad (3)$$

Рассмотрим набор переменных \mathbf{r} . Вынося r_1 и r_2 как общие множители и переписывая систему (3) в матричном виде, получаем:

$$\underbrace{\begin{pmatrix} a_{11}s_1 + a_{12}s_2 & a_{13}s_1 + a_{14}s_2 \\ a_{21}s_1 + a_{22}s_2 & a_{23}s_1 + a_{24}s_2 \\ a_{31}s_1 + a_{32}s_2 & a_{33}s_1 + a_{34}s_2 \end{pmatrix}}_B \times \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (4)$$

Ясно, что при фиксированных значениях \mathbf{s} эта система будет иметь нетривиальные по \mathbf{r} решения, если ранг матрицы B не полон (в противном случае система имеет единственное решение – $\mathbf{r} = \mathbf{0}$) [2]. Составим из трёх миноров второго порядка матрицы B новую систему уравнений:

$$\begin{cases} M_1 = Q_1(\mathbf{s}) = 0, \\ M_2 = Q_2(\mathbf{s}) = 0, \\ M_3 = Q_3(\mathbf{s}) = 0. \end{cases} \quad (5)$$

где $Q_i(\mathbf{s})$ – квадратичные формы от \mathbf{s} . Эту систему можно переписать в матричном виде, выделяя в качестве вектора неизвестных мономы второго порядка относительно \mathbf{s} :

$$M \times \begin{pmatrix} s_1s_2 \\ s_1^2 \\ s_2^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (6)$$

Рассмотрим случай, когда ранг матрицы M полон, т. е. матрица M невырождена. Тогда система (6) имеет единственное решение [2]. Как следствие, $\mathbf{s} = \mathbf{0}$. Это означает, что система (3) имеет только тривиальные решения. Отсюда вытекает

Теорема 1. (достаточное условие регулярности): если определитель матрицы M не равен нулю, то билинейная $(3;2,2)$ -система является регулярной.

Следствие 1. (необходимое условие нерегулярности): если билинейная $(3;2,2)$ -система является нерегулярной, то определитель матрицы M равен нулю.

Определитель матрицы M имеет шестой порядок, зависит от двенадцати коэффициентов и выглядит довольно громоздко. Пусть $d_{i,j,k}$ – минор второго порядка в обобщенной матрице системы (3), полученный вычеркиванием i -й строки и столбцов j и k . Тогда определитель матрицы M будет иметь вид:

$$\begin{aligned} \det M = & (d_{2,1,4}d_{3,2,4} + d_{2,2,3}d_{3,2,4} - d_{2,2,4}d_{3,1,4} - d_{2,2,4}d_{3,2,3})d_{1,1,3} + \\ & + (d_{1,2,4}d_{3,1,4} + d_{1,2,4}d_{3,2,3} - d_{1,1,4}d_{3,2,4} - d_{1,2,3}d_{3,2,4})d_{2,1,3} + \\ & + (d_{1,1,4}d_{3,2,4} + d_{1,2,3}d_{2,2,4} - d_{1,2,4}d_{2,1,4} - d_{1,2,4}d_{2,2,3})d_{3,1,3}. \end{aligned}$$

Замечание. Аналогичные рассуждения относительно набора переменных s приводят к точно такому же определителю.

3. Критерий регулярности для $(3;2,2)$ -систем

Перепишем систему (3) в виде (2):

$$\begin{cases} \mathbf{r}^T \cdot A_1 \cdot \mathbf{s} = 0 \\ \mathbf{r}^T \cdot A_2 \cdot \mathbf{s} = 0 \\ \mathbf{r}^T \cdot A_3 \cdot \mathbf{s} = 0 \end{cases} \quad (7)$$

Обозначим *допустимые преобразования*, которые не изменяют факта регулярности произвольной $(l; m, n)$ -системы:

- гауссовы преобразования уравнений в системе,
- замены переменных $\mathbf{r} = C \cdot \mathbf{r}^*$, $\mathbf{s} = D \cdot \mathbf{s}^*$ (матрицы C и D невырожденные), изменяющие матрицу билинейной формы A_i по закону:

$$A_i^* = C^T \cdot A_i \cdot D.$$

Для дальнейших преобразований потребуется хотя бы одно уравнение с невырожденной матрицей билинейной формы.

Лемма 1. Если $(3;2,2)$ -система является системой полного ранга, то существует гауссово преобразование билинейных форм из этой системы, приводящее к системе с невырожденной матрицей билинейной формы.

Используя эту лемму, можно получить систему с одной невырожденной матрицей билинейной формы. Пусть этой матрицей будет

матрица A_1 . Тогда «улучшим» вид этой матрицы, вводя замену $\mathbf{r} = A_1^T \cdot \mathbf{r}^*$, $\mathbf{s} = \mathbf{s}^*$. В результате получаем

$$A_1^* = (A_1^T)^T \cdot A_1 \cdot E = E.$$

Произведём ещё одну замену так, чтобы «улучшить» вид формы A_2^* , полученной после первой замены, сохраняя тривиальный (единичный) вид формы A_1^* . Вид матрицы A_2' будет зависеть от собственных значений и собственных векторов матрицы A_2^* . Для матрицы возможны следующие три случая [3]:

- различные собственные значения (простой спектр);
- кратные собственные значения;
- комплексные собственные значения.

Рассматривая все три случая, можно сформулировать промежуточную теорему об упрощенном виде системы (7):

Теорема 2. *Любая билинейная (3;2,2) -система полного ранга приводима допустимыми преобразованиями к одному из трёх упрощенных видов:*

$$\begin{aligned}
 1) \begin{cases} r_1 s_1 + r_2 s_2 = 0 \\ r_1 s_2 - r_2 s_1 = 0 \\ a'_1 r_1 s_1 + a'_2 r_2 s_1 = 0 \end{cases} & \quad 2) \begin{cases} r_1 s_1 = 0 \\ r_1 s_1 + r_2 s_2 = 0 \\ a'_3 r_1 s_1 + a'_4 r_2 s_1 = 0 \end{cases} \\
 3) \begin{cases} r_1 s_1 = 0 \\ r_2 s_2 = 0 \\ r_1 s_2 \cos \varphi + r_2 s_1 \sin \varphi = 0 \end{cases} & \quad (8)
 \end{aligned}$$

где a'_1, a'_2, a'_3, a'_4 и φ – некоторые коэффициенты, причём в парах a'_1, a'_2 и a'_3, a'_4 их элементы не могут быть одновременно нулевыми.

Рассмотрим условия, при которых упрощенные виды (8) могут иметь нетривиальные решения. Воспользуемся необходимым условием нерегулярности, которое сформулировано в следствии 1 и вычислим определители соответствующих матриц для каждого вида, приравнявая их к нулю:

$$\begin{aligned}
 1) \det M = -(a_1'^2 + a_2'^2) = 0, \quad 2) \det M = a_4' = 0, \\
 3) \det M = \cos \varphi \cdot \sin \varphi = 0. \quad (9)
 \end{aligned}$$

Накладывая ограничения (9) на коэффициенты в упрощенных формах (8), мы получим *канонические* виды, которые имеют нетривиальные решения. Отсюда формулируется необходимое и достаточное условие

нерегулярности для билинейной системы полного ранга (критерий нерегулярности):

Теорема 3. (критерий регулярности): билинейная $(3;2,2)$ -система полного ранга регулярна тогда и только тогда, когда она не приводима к одному из двух канонических видов:

$$1) \begin{cases} r_1 s_1 = 0 \\ r_2 s_2 = 0 \\ r_1 s_2 = 0 \end{cases}, \quad 2) \begin{cases} r_1 s_1 = 0 \\ r_2 s_2 = 0 \\ r_2 s_1 = 0 \end{cases}$$

Заключение

В данной работе была рассмотрена проблема определения нерегулярности однородной системы билинейных уравнений. В общем случае эта задача слишком сложна для рассмотрения, поэтому были исследованы более простые случаи. Досконально был проанализирован случай $(3;2,2)$ -системы и был выведен критерий регулярности для случая системы полного ранга, который позволяет, не решая систему, определить факт её регулярности. Критерий также работает и в сторону нерегулярности – для этого достаточно инвертировать формулировку теоремы 3.

Для случая $(3;2,2)$ -систем также было представлено достаточное условие регулярности, которое по виду системы даёт возможность в некоторых случаях однозначно установить регулярность билинейной системы.

Список литературы

1. Cohen S. Systems of bilinear equations. / S. Cohen, C. Tomasi // Technical Report, Stanford University Computer Science Department. – Stanford, CA, USA, 1997. – 40 p.
2. Кострикин А. И. Введение в алгебру. Часть I. Основы алгебры : Учебник для вузов. / А. И. Кострикин – 2-е изд. – М. : Физматлит, 2000. – 272 с.
3. Кострикин А. И.: Введение в алгебру. Часть II. Линейная алгебра : Учебник для вузов. / А. И. Кострикин. – 2-е изд. – М. : Физматлит, 2000. – 368 с.

ИНТЕГРИРОВАНИЕ МАТРИЧНЫХ АЛГЕБР ЛИ

В.А. Назарьева

Студент

Т.В. Новицкая

Студент

А.В. Лобода

Профессор

Введение

В данной работе представлено решение задачи интегрирования одного семейства матричных алгебр Ли (полученного Е. Павловой в [1]). Эта задача является частью объемного исследования, связанного с описанием аффинно-однородных поверхностей 3-мерного комплексного пространства [2, 3]. Изучаемое семейство алгебр описывает поверхности г. н. типа $(1/2, \varepsilon)$.

Работа выполнена в рамках изучения курса «Математические методы в современных информационных технологиях».

1. Постановка задачи

Необходимо найти функцию комплексных переменных $\Phi = -v + F(z_1, z_2, u)$, где $z_1 = x_1 + iy_1$, $z_2 = x_2 + iy_2$, $w = u + iv$, которая должна удовлетворять уравнениям:

$$\Re\{E_k(\Phi)\}\big|_{\Phi=0}, \quad k = 1 \dots 5. \quad (1)$$

Здесь каждое из 5 базисных векторных полей E_k имеет следующий вид:

$$\begin{aligned} & (A_1 z_1 + A_2 z_2 + A_3 w + p) \frac{\partial}{\partial z_1} + (B_1 z_1 + B_2 z_2 + B_3 w + s) \frac{\partial}{\partial z_2} + \\ & + (a z_1 + b z_2 + c w + q) \frac{\partial}{\partial w}. \end{aligned} \quad (2)$$

В упрощенной форме алгебры аффинных векторных полей можно рассматривать как матричные алгебры, состоящие из матриц вида:

$$\begin{bmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

Система (1) имеет решение, если базисные матрицы (3) векторных полей (2) образуют матричную алгебру Ли (линейное пространство матриц, замкнутое относительно коммутатора) [4].

Базисные матрицы в нашем случае зависят от двух вещественных параметров (ε , t_{31}) и имеют следующий конкретный вид:

$$E_1 = \begin{bmatrix} 0 & -\frac{(-4\varepsilon + 4I\varepsilon - 2 - 2I)t_{31}}{-1 + 4\varepsilon^2} & -\frac{2It_{31}^2(4\varepsilon^2 + 1)}{(1 + 4\varepsilon + 4\varepsilon^2)(-1 + 2\varepsilon)^2} & 1 \\ 0 & 0 & 0 & 0 \\ 4I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$E_2 = \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_3 = \begin{bmatrix} \frac{2t_{31}}{-1 + 2\varepsilon} & 0 & 0 & 0 \\ 0 & (-1 - I)t_{31} & 0 & 1 \\ 0 & 2I(I + 2\varepsilon) & \frac{2t_{31}}{-1 + 2\varepsilon} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$E_4 = \begin{bmatrix} \frac{2t_{31}}{1 + 2\varepsilon} & 0 & 0 & 0 \\ 0 & (1 - I)t_{31} & 0 & I \\ 0 & 2 - 4\varepsilon & \frac{2t_{31}}{1 + 2\varepsilon} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (4)$$

где $0 < \varepsilon \neq \frac{1}{2}$, $t_{31} \neq 0$.

Вещественное 5-мерное линейное пространство матриц с базисом (4) является матричной алгеброй Ли, так как коммутатор любых двух базисных матриц из (4) может быть представлен в виде линейной комбинации этих же пяти матриц.

2. Интегрирование алгебр

Для интегрирования любой из матричных алгебр Ли (4) достаточно решить систему (1). Однако базис (4) исходной алгебры устроен достаточно сложно, поэтому необходимо сначала преобразовать базисные поля. Кроме того, систему (1) нужно записать в вещественных координатах.

Для упрощения вводим подобие типа $E_{1m} = C * E_{1s} * C^{-1}$ с матрицей:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\frac{1}{2}It_{31}(2\varepsilon+1-I+2I\varepsilon)}{-1+4\varepsilon^2} & 0 & \frac{\frac{1}{2}-\frac{1}{2}I}{t_{31}} \\ 0 & 1 & 1 & \frac{\left(-\frac{1}{2}-\frac{1}{2}I\right)(-1+4\varepsilon^2)}{t_{31}^2} \\ 0 & 0 & 0 & \frac{1}{1} \end{bmatrix}, \quad (5)$$

где S – это матрица составленная из собственных векторов E_4 .

В результате подобия получаем новый улучшенный базис (6). Полученные в результате подобия поверхности сводятся аффинными преобразованиями к исходным поверхностям.

$$E_{1m} = \begin{bmatrix} 0 & 0 & IA & IB \\ 0 & 0 & 0 & 0 \\ 4I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_{2m} = \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$E_{3m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2\varepsilon - I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_{4m} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6)$$

$$E_{5m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

где $A = -\frac{2t_{31}^2(4\varepsilon^2+1)}{(4\varepsilon^2+4\varepsilon+1)(-1+2\varepsilon)^2}$, $B = \frac{(-1+2\varepsilon)}{(1+2\varepsilon)}$, $\mu = -4\varepsilon^2+1$.

Выписываем систему комплексных уравнений в частных производных, описанных данным базисом и формулой (1):

$$\Re\left\{ (IwA + IB) \frac{\partial F}{\partial z_1} + 4Iz_1 \frac{\partial F}{\partial w} \right\} = 0, \quad \Re\left\{ I \frac{\partial F}{\partial z_1} \right\} = 0,$$

$$\Re\left\{ z_2(2\varepsilon - I) \frac{\partial F}{\partial z_2} \right\} = 0, \quad (7)$$

$$\Re \left\{ \mu z_2 \frac{\partial F}{\partial z_2} + 2w \frac{\partial F}{\partial w} + 2z_1 \frac{\partial F}{\partial z_1} \right\} = 0, \quad \Re \left\{ \frac{\partial F}{\partial w} \right\} = 0.$$

Используя свойства дифференцирования функций комплексной переменной:

$$\frac{\partial F}{\partial z_1} = \frac{1}{2} \left(\frac{\partial F}{\partial x_1} - i \frac{\partial F}{\partial y_1} \right), \quad \frac{\partial F}{\partial z_2} = \frac{1}{2} \left(\frac{\partial F}{\partial x_2} - i \frac{\partial F}{\partial y_2} \right), \quad \frac{\partial F}{\partial w} = \frac{1}{2} \left(\frac{\partial F}{\partial u} - i \frac{\partial F}{\partial v} \right) \quad (8)$$

и условие $v = F$, отвечающее равенству $\Phi = 0$, перейдем к вещественной форме уравнений:

$$\begin{aligned} U_1 : -\frac{AF}{2} \left(\frac{\partial F}{\partial x_1} \right) - 2x_1 = 0, \quad U_2 : \frac{\partial F}{\partial y_1} = 0, \\ U_3 : \left(\epsilon x_2 + \frac{1}{2} y_2 \right) \frac{\partial F}{\partial x_2} - \left(\frac{1}{2} x_2 - \epsilon y_2 \right) \frac{\partial F}{\partial y_2} = 0, \\ U_4 : \frac{1}{2} \mu x_2 \frac{\partial F}{\partial x_2} + \frac{1}{2} \mu y_2 \frac{\partial F}{\partial y_2} - F + x_1 \frac{\partial F}{\partial x_1} = 0, \quad U_5 : \frac{\partial F}{\partial u} = 0. \end{aligned} \quad (9)$$

В этой системе функция F зависит от переменных x_1, x_2, y_2 и не зависит от y_1, u , что показано в уравнениях U_2, U_5 .

Для нахождения решения полной системы (9) будем поочередно решать ее отдельные уравнения. Любое из них имеет следующий (общий) вид квазилинейного уравнения в частных производных:

$$A_1 \frac{\partial F}{\partial x_1} + A_2 \frac{\partial F}{\partial x_2} + B_1 \frac{\partial F}{\partial y_1} + B_2 \frac{\partial F}{\partial y_2} + C \frac{\partial F}{\partial u} = D. \quad (10)$$

Решая по известным схемам [5] уравнение U_1 , получаем:

$$F = \frac{\sqrt{H(x_2, y_2) - 4Ax_1^2}}{A}, \quad (11)$$

где $H(x_2, y_2)$ – произвольная аналитическая функция двух переменных.

Подставляя $F(x_1, x_2, y_2)$ в U_4 и решая полученное уравнение, получаем:

$$H(x_2, y_2) = x_2 \left(\frac{4}{\mu} \right) G \left(\frac{y_2}{x_2} \right), \quad (12)$$

где $G \left(\frac{y_2}{x_2} \right)$ – произвольная аналитическая функция одной переменной.

Далее вводим замену $s = y_2 / x_2$. Подставляя $H(x_2, y_2)$ в U_4 и решая полученное уравнение, получаем:

$$G(s) = C(s^2 + 1)^{\frac{2}{\mu}} e^{\frac{8\varepsilon \arctan(s)}{\mu}}, \quad (13)$$

где C – произвольная константа.

С учетом всех замен и подстановок, описанных выше, искомая функция F примет вид:

$$F(x_1, x_2, y_2) = \frac{\sqrt{x_2 \left(\frac{4}{\mu}\right) C \left(\left(\frac{y_2}{x_2} \right)^2 + 1 \right)^{\frac{2}{\mu}} e^{\frac{8\varepsilon \arctan\left(\frac{y_2}{x_2}\right)}{\mu}} - 4Ax_1^2}{A}. \quad (14)$$

Полученное уравнение поверхности $v = F(x_1, x_2, y_2)$ можно упростить, выполнив несложные преобразования и введя замены $\alpha = 2/\mu$, $\beta = 8\varepsilon/\mu$.

В итоге уравнение любой из поверхностей изучаемого семейства принимает (с точностью до аффинных преобразований) окончательный вид:

$$v^2 + x_1^2 = |z_2|^\alpha e^{\beta \arg(z_2)}. \quad (15)$$

Отметим, что операции с матрицами и решение отдельных уравнений в частных производных осуществлялись в данной работе с использованием математического пакета Maple 14, в котором также была выполнена проверка получившегося решения. При подстановке полученного уравнения искомой поверхности (14) в каждое уравнение первоначальной системы, получаются верные тождества.

Заключение

Данная статья посвящена решению задачи интегрирования матричных алгебр Ли. В ходе данной работы было проинтегрировано зависящее от двух вещественных параметров семейство матричных алгебр Ли и получены уравнения соответствующих гиперповерхностей 3-мерного комплексного пространства.

Литература

1. Павлова, Е.А. Компьютерные алгоритмы описания аффинно-однородных поверхностей / Е.А. Павлова; под ред. Д.Н. Борисова // Сборник студенческих работ факультета компьютерных наук. – 2015. – Вып. 9. – С.105-113.
2. Лобода, А.В. Аффинно-однородные вещественные гиперповерхности 3-мерного комплексного пространства / А.В. Лобода // Вестник ВГУ. Серия : физика, математика. – 2009. – С. 71-91.
3. Лобода, А.В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства /

А.В. Лобода, А.С. Ходарев // Известия ВУЗов. Математика. – 2003. – №10. – С. 38-50.

4. Кочурова, Е.Д. Интегрирование матричных алгебр Ли / Е.Д. Кочурова, А.В. Лобода // Сборник студенческих работ факультета компьютерных наук. – 2013. – Вып. 7. – С.68-76.

5. Понтрягин, Л.С. Обыкновенные дифференциальные уравнения / Л.С. Понтрягин. – М. : Наука, 1974. – 331 с.

ТЕКСТОНЕЗАВИСИМАЯ ИДЕНТИФИКАЦИЯ ДИКТОРА ПО ГОЛОСУ

А.А. Ногих

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Способности человека к распознаванию других людей по голосу являются предметом научных исследований уже более 80 лет. Устанавливая факторы, различающие голоса разных людей, исследователи смогли добиться значительных успехов в синтезировании голоса и его обработке. С появлением цифровых компьютеров разработка технологий для автоматического распознавания диктора перешла в практическую область. Было обнаружено, что в этой конкретной задаче компьютерные системы способны превосходить человека по точности [1].

Интерес к проблеме распознавания диктора продолжает расти. Это можно объяснить широкими возможностями применения данной технологии:

- Криминалистика и судебная экспертиза;
- Безопасность;
- Банковские технологии;
- Электронная коммерция.

Задача распознавания диктора разделяется на две подзадачи:

- Идентификация – установление принадлежности записи голоса одному из известных системе распознавания дикторов;
- Верификация – установление принадлежности данной записи определенному диктору.

Также системы распознавания диктора могут быть текстозависимыми и текстонезависимыми. В первом случае ограничено текстовое

содержимое высказываний, поступающих на обработку, во втором – системы являются более общими.

В данной работе рассматривается задача текстонезависимой идентификации диктора по голосу на основе мэл-частотных кепстральных коэффициентов и векторного квантования. Задача включает в себя реализацию и исследование параметров реализованного алгоритма.

1. Процесс идентификации диктора

Процесс идентификации диктора включает в себя два этапа:

– Этап обучения (рис. 1) – формируется набор эталонных моделей дикторов.

– Процесс идентификации (рис. 2) – применяются полученные на этапе обучения модели дикторов. Происходит принятие решения на основе этих моделей и поступающей голосовой информации.

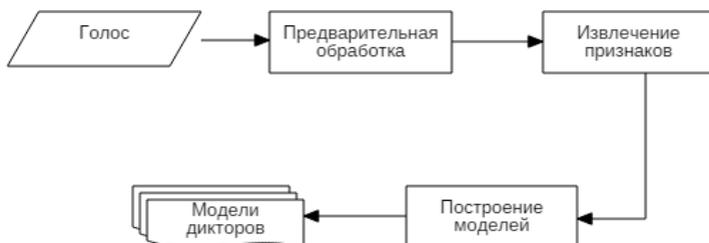


Рис. 1. Схема работы в режиме обучения

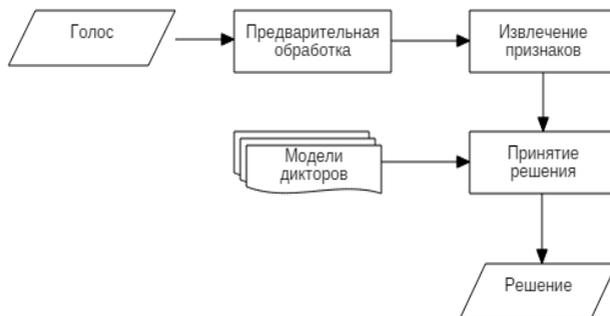


Рис. 2. Схема работы в режиме идентификации

2. Предварительная обработка и извлечение признаков

Голос поступает на вход закодированным с использованием импульсно-кодовой модуляции. На данном этапе есть возможность

применить фильтры, делающие сигнал более пригодным к дальнейшей обработке. В данной работе был применен pre-emphasis filtering, усиливающий необходимые для определения диктора частоты звука [2].

Входной сигнал разбивается на временные окна (фреймы), на которых, как предполагается, параметры сигнала значительно не изменяются. Для более точного представления сигнала между окнами делают перекрытие, равное половине длины окна. В данной работе используются перекрывающиеся фреймы длиной 30 миллисекунд.

После предварительной обработки из сигнала извлекаются значимые для распознавания диктора характеристики. Для гласоподобных звуков одной из лучших характеристик считается набор формантных частот (формант). Форманты – проявления резонансных частот речевого тракта диктора в акустическом сигнале. Большинство систем автоматического распознавания диктора основываются именно на анализе формантной картины [3].

Следует также учесть факт того, что слуховое восприятие звука человеком оценивается не частотой звука, а высотой тона. Единицей высоты тона определен Мэл – «внесистемная единица высоты звука», количественная оценка высоты тонов основывается на статистической обработке данных о субъективном восприятии звука [4].

Подход, основанный на использовании мэл-частотных кепстральных коэффициентов, выглядит описанным далее образом [2, 5].

На входе – фрейм, представленный набором отсчетов $s(n)$. Применяется весовая функция $w(n)$ с целью подавления побочных спектров в последующем преобразовании Фурье:

$$X_k = \sum_{n=0}^{N-1} s(n)w(n)e^{-\frac{2\pi i}{N}kn}. \quad (1)$$

Полученное частотное представление сигнала разбивают на диапазоны с помощью банка фильтров, границы которых рассчитаны в шкале мэл. Выбиваются весовые функции φ_k , которые будут фильтровать заданные диапазоны. Они могут иметь треугольный вид:

$$\varphi_k(n) = \begin{cases} \frac{n - \text{left}}{\text{avg} - \text{left}}, & n < \text{avg} \\ 1, & n = \text{avg} \\ \frac{\text{right} - n}{\text{right} - \text{avg}}, & n > \text{avg} \end{cases} \quad (2)$$

где left и right – левая правая границы диапазона соответственно, avg – центральная точка диапазона. Еще одним вариантом является применение весовых функций, основанных на распределении Гаусса:

$$\varphi(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-\mu)^2}{2\sigma^2}}, \quad (3)$$

где $\sigma = \frac{\text{right-avg}}{2}$, $\mu = \text{avg}$.

Полученные значения суммируются и логарифмируются:

$$e_n = \log \left[\sum_k^{N-1} |X_k|^2 \varphi_n(k) \right], n = 0, 1, \dots, N_f - 1. \quad (4)$$

Заключительным этапом получения мэл-частотных кепстральных коэффициентов c_n является вычисление дискретно-косинусного преобразования, выходные значения которого и образуют вектор признаков.

Вектор признаков можно расширить, вычисляя дельта-значения:

$$d_t = \frac{\sum_{n=1}^N n(c_{n+t} - c_{n-t})}{2 \sum_{n=1}^N n^2}. \quad (5)$$

3. Построение модели и принятие решения

Векторное квантование – процесс отображения большого количества векторов в конечное число областей векторного пространства. Каждая такая область называется кластером и может быть представлена своим центроидом, называемым кодовым вектором. Множество кодовых слов для одного диктора называется кодовой книгой [6]. Кодовая книга $C = \{c_n\}$ будет полностью определять модель диктора.

В данной работе был применен алгоритм Линдо-Бузо-Грея (подробнее см. [6]) для выполнения кластеризации.

Определим $\theta : C \times x \rightarrow R$. как функцию, возвращающую минимальное расстояние от вектора признаков до элементов кодового словаря: $\min d(c_i, x)$, $i = 1, 2, \dots, |C|$. Тогда для каждого отдельного вектора признаков x диктор определяется как:

$$\lambda_x = \arg \min \theta(C_i, x). \quad (6)$$

Пусть $t(\lambda_i)$ – количество векторов, по формуле выше отнесенных к модели диктора λ_i . Окончательное решение тогда примет вид:

$$\lambda_{res} = \arg \max t(\lambda_i). \quad (7)$$

4. Задача и методика эксперимента

В рассматриваемом подходе к идентификации диктора есть параметры, которые необходимо установить экспериментально:

- Размер используемого банка мэл-частотных фильтров;
- Количество мэл-кепстральных коэффициентов (MFCC);

- Вид фильтрующей функции для банка фильтров;
- Необходимость учета дельта-значений MFCC.

В качестве голосовых материалов была взята находящаяся в открытом доступе на сайте университета Карнеги-Мэллон (США) база записей. Получить ее можно по адресу <http://www.speech.cs.cmu.edu/databases/an4/>, перейдя по ссылке «Raw audio (.raw) format, big endian byte order».

В ней содержатся закодированные в формате PCM аудиофайлы с частотой дискретизации 16 КГц (разрядность АЦП 16 бит). База содержит записи голосов 74 человек (53 диктора-мужчины и 24 диктора-женщины).

Для обучения и тестирования использовались разные высказывания. Средняя длина обучающей выборки – 16 секунд на каждый голос, выборки, отведенной на тестирование – 6 с каждого голоса. Для тестирования все тестовые высказывания были отделены и разбиты на фрагменты длиной 0,32 секунды каждый (20 векторов признаков при использовании перекрывающихся фреймов длиной 512 отсчетов).

При получении результатов использовались 30 случайных выборок по 16 дикторов. Процент точности в каждой выборке вычислялся как отношение правильно распознанных фрагментов речи ко всем, участвовавшим в распознавании.

Итоговый процент точности получался как среднее значение между результатами каждой из случайных выборок.

5. Результаты эксперимента

Измерения проводились при следующих условиях:

- Размер кодовой книги: 128;
- Форма фильтрующих функций: гауссовая (3);
- Вид вектора признаков: мэл-частотные кепстральные коэффициенты + дельта-значения (5)

Как видно из таблицы 1, наибольшая эффективность (89,20%) достигается при использовании банка фильтров размером в 40 и извлечением из него 28 MFCC. В последующих экспериментах за основу будут браться именно эти значения.

Далее был определен вид вектора признаков (таблица 2). DMFCC означает добавление к вектору признаков дельта-значений MFCC к вектору признаков, а DDMFCC – добавление дельта-дельта-значений MFCC (получаемых путём применения формулы 5 дважды).

Эксперимент показал, что применение гауссовых фильтрующих функций (3) приводит к большей точности распознавания, чем применение треугольных (2). В плане учета дельта-коэффициентов были показаны примерно одинаковые результаты. Самую большую точность (89,20%) дало применение векторов признаков, состоящих из MFCC и DMFCC.

Также была измерена зависимость эффективности реализованного решения при определенных выше оптимальных параметрах от количества дикторов. Эта зависимость изображена на рис. 3.

Таблица 1

Результаты оценки размера банка мэл-частотных фильтров и количества мэл-частотных кепстральных коэффициентов (MFCC)

		Банк фильтров					
		20	25	30	35	40	45
MFCC	10	74,91%	76,76%	77,63%	79,58%	80,11%	80,11%
	12	78,00%	79,68%	80,51%	82,34%	82,68%	83,82%
	14	78,04%	81,30%	83,43%	84,47%	84,16%	86,08%
	16	77,78%	82,40%	83,31%	85,83%	85,88%	85,54%
	18	78,05%	82,60%	84,94%	86,74%	86,72%	86,92%
	20	78,94%	82,55%	85,84%	86,80%	87,53%	87,96%
	22	79,19%	82,86%	85,46%	87,38%	88,02%	88,17%
	24	79,73%	83,47%	86,08%	87,80%	88,43%	88,49%
	26	81,05%	83,15%	85,57%	87,77%	88,60%	88,32%
	28	82,61%	85,28%	86,21%	88,52%	89,20%	89,10%
	30	83,55%	84,88%	85,48%	87,99%	88,02%	89,08%

Таблица 2

Результаты оценки необходимости учета дельта-коэффициентов и вида используемых фильтров

Учет дельта-коэффициентов	Треугольные	Гауссовы
MFCC	81,64%	88,41%
MFCC+DMFCC	82,79%	89,20%
MFCC+DMFCC+DDMFCC	81,72%	89,11%

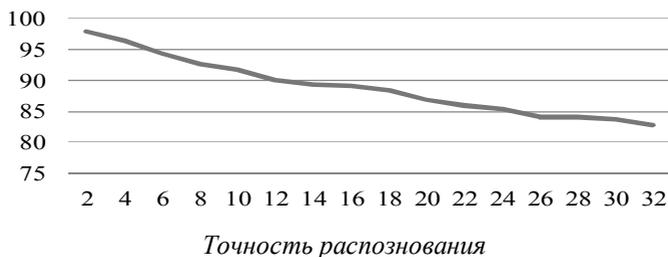


Рис. 3. Результаты оценки зависимости точности распознавания от количества дикторов

Заключение

В рамках данной работы был исследован и реализован метод текстонезависимой идентификации дикторов, основанный на применении мел-частотных кепстральных коэффициентов и векторного квантования.

Был проведен эксперимент, позволивший установить параметры алгоритма, приводящие к наибольшей точности идентификации, а также его оценить поведение при росте количества альтернативных вариантов.

В качестве дальнейшего направления работы можно рассмотреть задачу верификации диктора и исследование применения альтернативных алгоритмов, используемых при решении данной задачи.

Список использованной литературы

1. Reynolds Douglas A.. Automatic Speaker Recognition Using Gaussian Mixture Speaker Models / Douglas A. Reynolds // The Lincoln Laboratory Journal. – 1995. – Vol. 8. – № 2. – P. 173-177.
2. Lindsalwa Muda. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques / Muda Lindsalwa, Mumtaj Begam, I. Elamvazuthi // Journal of Computing. – 2010. – Vol. 2. – Issue 3. – P. 138-143.
3. Вишнякова, О. А. Подходы к задаче идентификации диктора / О. А. Вишнякова, Д. Н. Лавров // Математические структуры и моделирование. – 2011. – Вып. 22. – С. 48-54.
4. Русаков, И. Г. Мел (единица высоты звука) / И. Г. Русаков // Большая Советская Энциклопедия : в 30-и т. Т. 16. – 3-е изд. – М. : Издательство «Советская энциклопедия», 1974. – 136 с.
5. Ермилов, А. В. Методы, алгоритмы и программы решения задач идентификации языка и диктора : дисс. ... канд. физ.-мат. наук : 05.13.11 / А. В. Ермилов. – М. : ВШЭ, 2011. – 135 с.
6. Ахмад, Х. М. Исследование и разработка алгоритмов параметризации речевых сигналов в системе распознавания диктора : дисс. ... канд. тех. наук : 05.13.01 / Ахмад Хассан Мухаммад – Владимир : ВлГУ, 2008. – 157 с.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ КОНТРОЛЯ АУТЕНТИЧНОСТИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

Е.Ю. Обущенко

Студент

М.А. Дрюченко

Доцент

Введение

На сегодняшний день информация является одной из ценнейших вещей в жизни людей и требует разработки современных механизмов для защиты от несанкционированного доступа, нарушения конфиденциальности и целостности. В 60-х – 70-х годах проблема защиты информации решалась достаточно эффективно, поскольку применялись организационные меры, к которым, в первую очередь, относились режимные мероприятия, охрана, сигнализация и простейшие программные средства защиты информации. Но наибольшая эффективность использования данных средств достигалась с помощью концентрации информации на вычислительных центрах, большая часть из которых была автономной, что помогало обеспечить защиту относительно малыми средствами.

С появлением большого количества недорогих персональных компьютеров и построенные на их основе локальных и глобальных сетей ЭВМ, которые используют спутниковые каналы связи, появление эффективных систем разведки и поиска информации, было вызвано распространение информации по местам её хранения и обработки, что обострило положение защиты информации.

Именно поэтому очень остро встала проблема обеспечения необходимого уровня защиты информации, так как последнее время люди всё больше и больше обмениваются сканированными документами, фотографиями не только общедоступными, но и личной или юридической значимости, которые требуют максимального уровня защиты. Так для повышения безопасности данных файлов от несанкционированных изменений следует разработать математическое и программное обеспечение для контроля аутентичности цифровых изображений с возможностью эффективного обнаружения и локализации областей подверженных модификации.

1. Анализ известных методов контроля целостности и аутентичности цифровых данных

В настоящее время разработано множество методов для контроля целостности и аутентичности цифровых данных, одними из самых известных и популярных являются: метод контрольных сумм, хеширование, электронная подпись и цифровой водяной знак. Подробнее рассмотрим каждый из них.

Метод контрольных сумм. Контрольная сумма – это некоторое значение, которое рассчитывается по набору данных путём применения определённого алгоритма и используется для проверки целостности данных при их передаче или хранении. Этот метод имеет свои несомненные плюсы, одни из них – это простота реализации и высокая производительность. Но, в свою очередь, равенство полученных сумм не даёт гарантии неизменности информации, а также не составит труда подобрать данные так, что бы получилось правильное значение.

Следующий метод контроля – хеширование. Это преобразование по определённому алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования ещё называют хеш-функциями или «функцией свертки», а результаты называют хешем, хеш-кодом, хеш-суммой или «сводкой сообщения» [1]. Именно использование данного метода даёт ряд преимуществ, таких как вычислительная сложность, совместимость и целостность.

Так же следует вспомнить про электронную подпись, которая представляет собой некий реквизит электронного документа, полученного в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи, а так же принадлежность подписи владельцу сертификата ключа [1]. В случае же успешной проверки подтвердить факт подписания электронного документа.

И последний метод, про который необходимо отметить – это цифровой водяной знак (ЦВЗ), или специальная метка, которая незаметно внедряется в изображение с целью определённым образом контролировать его использование. Наряду с видимыми ЦВЗ бывают так же и невидимые, которые анализируются специальным декодером, который выносит решение об их корректности. ЦВЗ так же могут включать некоторую информацию о собственнике, аутентичный код или некую управляющую информацию. Неподвижные изображения, файлы видео и аудиоданные – являются одними из самых подходящих для защиты с помощью ЦВЗ.

Не смотря, на все свои плюсы, существующие методы контроля цифровых данных имеют ряд недостатков, характерных для каждого из них:

- чувствительность к малейшему изменению информации (например, сжатие изображения при сохранении в формате jpeg);
- отсутствие возможности локализовать вносимые изменения;
- коллизии (массивы данных с разным содержанием, но дающих одинаковые хеш-коды).

Для устранения этих недостатков была предложена комбинированная схема защиты цифровых данных.

2. Комбинированная схема контроля аутентичности цифровых изображений на основе использования стеганографических и криптографических методов

Проанализировав существующие алгоритмы контроля целостности и аутентичности цифровых изображений, а также известные алгоритмы стеганографического скрывания и алгоритмы выделения низкочастотной информативно-значимой составляющей цифровых изображений была предложена схема маркирования файла-изображения с целью контроля аутентичности, представленная на рис. 1.

На входе мы имеем файл-изображение, которое разбивается на блоки, для вычисления и стеганографического кодирования, с помощью стегоключа, перцептивного хеша на основе дискретного косинусного преобразования [2]. После мы получаем промежуточный маркированный файл. Далее вычисляем электронную подпись и записываем её в неиспользуемые области изображения. Получаем защищённый файл-изображение.

Для стегоскрывания использовался алгоритм Коха-Жао. Суть метода состоит в изменении отношения между абсолютными значениями коэффициентов ДКП в среднечастотной области изображения. Наименьшие искажения будут вноситься при модификации в ВЧ области, однако атака сжатием с использованием JPEG гарантировано разрушит встроенное сообщение. Наименьшая вероятность разрушения сообщения будет наблюдаться при встраивании в НЧ область, однако это приведет к появлению видимых искажений. В итоге на практике встраивание осуществляется в СЧ область [3]. Это условная область выбирается эмпирически как поиск компромисса между ожидаемым уровнем сжатия (при реализации соответствующей атаки) и требуемым качеством изображения.

Выделение низкочастотных информативно-значимых составляющих осуществлялось посредством алгоритма Canny. Он имеет пять основных этапов: сглаживание (удаление шума посредством сжатия изображения),

поиск градиентов (границы отмечаются там, где градиент изображения приобретает максимальное значение), подавление не-максимумов, двойная пороговая фильтрация, трассировка области неоднозначности (определяются итоговые границы, подавлением всех краёв, несвязанных с “сильными” границами) [4].

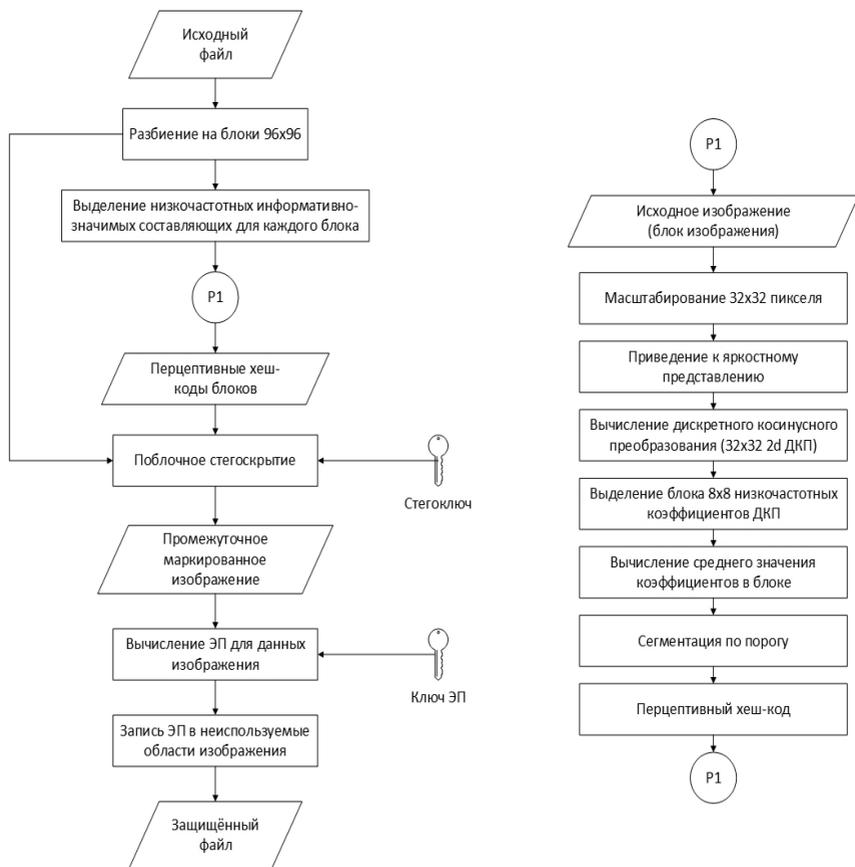


Рис. 1. Схема маркирования изображения с целью контроля аутентичности

Поскольку изображение мы маркировали, то следует проверить его на целостность и аутентичность, схема представлена на рис. 2.

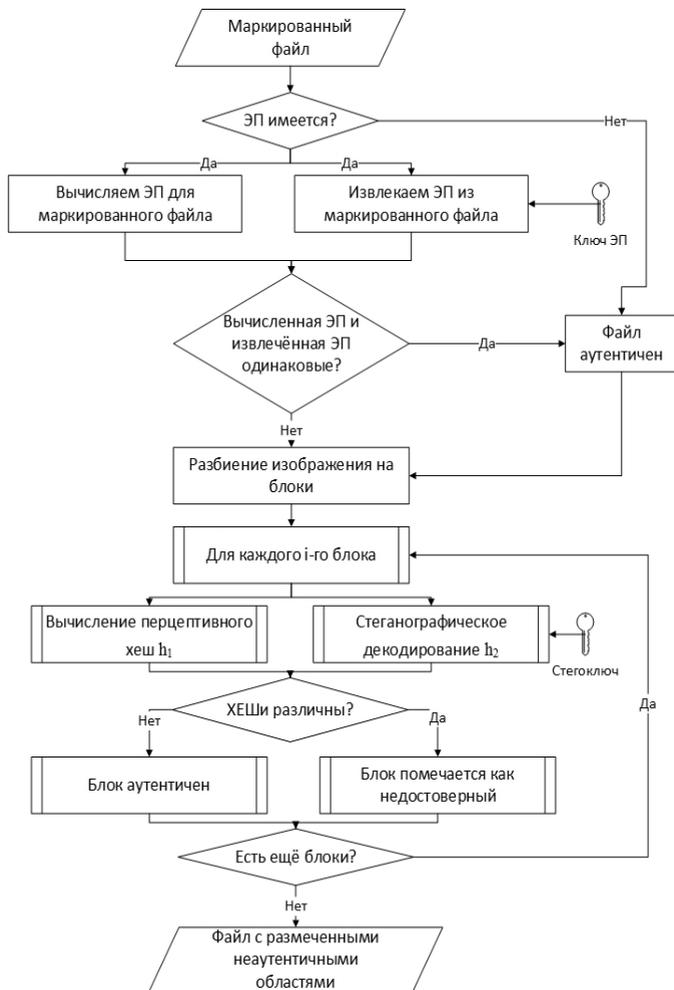


Рис. 2. Схема проверки аутентичности изображения

На входе мы получаем маркированный файл, после чего проверяется наличие электронной подписи. Если она имеется, то проверяется её подлинность. Далее изображение разбивается на блоки, и для каждого i -го блока вычисляется перцептивный хеш. Он сравнивается с стеганографически декодированным перцептивным хешем этого блока. Если хеш не совпадает, то блок помечается как неаутентичный. После

обработки всех блоков мы получаем файл-изображение с размеченными неаутентичными областями.

3. Тестирование и исследование эффективности предложенной схемы

Для тестирования предложенной схемы был создан программный модуль, написанный на языке программирования C#.

Были проанализированы 5 изображений фотографического качества и 5 файлов-изображений сканированных документов. Тестовые изображения подвергались маркировке, далее в каждое защищённое изображение вносились изменения. После реализовывалась проверка аутентичности на основе параметрически заданного порога для оценки меры близости вычисленных и стеганографически декодированных значений хеш-кодов блоков, принималось решение об аутентичности соответствующих блоков. Не прошедшие проверку блоки помечались как недостоверные. В итоге на выходе алгоритма формировалось изображение с размеченными областями, в которых возможно имели место модификации низкочастотного характера (информативно значимых составляющих).

Для оценки корректности работы алгоритмов проводилась визуальная оценка конечного результата в виде размеченных областей с учётом знания областей изначально подвергшихся модификации. Оценивались вероятности ошибки первого и второго рода, т.е. не обнаружения модифицированных фрагментов и обнаружения ложных модификаций.

Тестирование проводилось для файлов-изображений с разбиением на блоки размером 128x128, 96x96, 64x64 пикселей. При этом была выявлена следующая зависимость. При уменьшении размера блока количество ложных срабатываний алгоритма увеличивалось (из-за jpeg сжатия выходного изображения). В тоже время при увеличении размера блока точность обнаружения областей модификации снижалась. В результате проведенных экспериментов оптимальный размер блоков, обеспечивающий высокую точность обнаружения реальных областей модификации при минимальном количестве ложных срабатываний, составил 96x96 пикселей.

Заключение

После изучения и анализа существующих алгоритмов контроля целостности и аутентичности цифровых данных была предложена комбинированная схема аутентификации и контроля целостности цифровых изображений, использующая методы выделения низкочастотной информационно-значимой составляющей цифровых изображений, методы криптографии и методы полухрупкой стенографической маркировки цифровых данных.

Так же был разработан программный модуль для защиты цифровых изображений, которые позволяет максимально эффективно обнаруживать и локализовать возможные изменения защищенных данных.

Список литературы

1. Петров А. А. Компьютерная безопасность / А. А. Петров. — Криптографические методы защиты. — М. : ДМК. — 2000. — 448 с.
2. Стеганография : прячем данные в JPEG [Электронный ресурс] : электр. журн. — Режим доступа : <http://habrahabr.ru/post/206094/>
3. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. — М. : МК-Пресс, 2006. — 288 с.
4. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. — М. : Техносфера, 2005. — 1072 с.

РАЗРАБОТКА МОДЕЛЕЙ И АЛГОРИТМОВ ПРОСТРАНСТВЕННОЙ НАВИГАЦИИ ВНУТРИ ПОМЕЩЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ IBEAACON

А.Б. Ольферук

Студент

Н.А. Тюкачев

Доцент

Введение

В последнее время всё более актуальной становится проблема навигации внутри помещений, а также предоставления посетителям услуг, основанных на их местоположении и предпочтениях. Благодаря indoor-навигации (навигации внутри помещений) появляются новые инструменты для маркетинга: проходя мимо магазина, человек может моментально узнать о проводимых в нем акциях, мероприятиях, товарах и предоставляемых услугах, благодаря всплывающему сообщению на экране своего телефона (так называемом “Geo-fencing”, причём предложенные ему предложения будут учитывать его интересы, так как можно учитывать ин-формацию о его прошлых покупках), либо просто получить уведомление при приближении к определенному месту (второе направление indoor-навигации, называемое “Geo-aware”), а владельцы — получать статистическую информацию («тепловые карты» посетителей — своеобразный и очень мощный offline-аналог Google Analytics), основанные на перемещениях клиентов внутри торговых залов (таким

образом понять, какие отделы и товары пользуются повышенным интересом, очень легко). Рынок подобной геоконтекстной рекламы уже измеряется миллиардами долларов, и с развитием систем indoor-навигации ожидается его стремительный рост.

Bluetooth-маяки, представляющие собой один из способов решения проблемы локации, также являются частью Интернета вещей, и способны изменить представление о самом Интернете вещей. Google представила концепцию “physical web”, которая призвана объединить два мира: реальный и виртуальную Интернет-сеть. Девайс, сканируя ближайшие к нему маячки, получает из их сигнала встроенные URL-ссылки, доступные пользователю. Таким образом, пользователь может мгновенно получить доступ к информации, определенной в ближайшем локационном контексте. Кроме того, это шаг навстречу унифицированному, бесшовному интерфейсу взаимодействия с цифровым миром.

Но современный этап развития технологии внутренней навигации представлен в основном технологиями для навигации роботов по маякам. Кроме того, существует множество готовых продуктов и платформ от производителей с мировым именем. В то же время сообщество программистов не располагает подходящими материалами, чтобы оценить существующие варианты алгоритмов трилатерации и выбрать лучший способ в рамках собственных приложений или свободно распространяемыми библиотеками, которые можно было бы легко подстроить в рамках решаемой задачи.

Именно это и подтолкнуло меня к решению задач, поставленных в рамках настоящей работы.

1. Постановка задачи

Цель работы:

1. Проанализировать и сравнить различные методы трилатерации, используемые при решении задачи навигации внутри помещений:
 - 1.1. метод, основанный на поиске области пересечения сфер;
 - 1.2. метод, основанный на поиске силового центра;
 - 1.3. продвинутый геометрический алгоритм.
2. Разработать библиотеку, содержащую реализации вышеперечисленных алгоритмов.

К разрабатываемому проекту выдвинуты следующие требования:

3. Поддержка различных методов трилатерации;
4. Работа алгоритмов должна быть максимально оптимизирована как по скорости вычислений, так и по объему занимаемой памяти;
5. Первоначальная настройка библиотеки в проекте должна содержать минимум кода и быть интуитивно понятной.

2. Анализ задачи

В случае спутниковой навигации (GPS/Глонасс) существуют сервисы Out-Door, благодаря которым пользователь может узнать о ближайших кафе, ресторанах, гостиницах и других местах благодаря тому, что известно его текущее местоположение. А благодаря сервисам indoor-навигации возможно без проблем и оперативно найти ближайшую стойку регистрации в здании аэропорта, экспонат в музее, отдел и полку с нужным вам товаром в магазине, свободное место на парковке, и многое другое.

Благодаря большим коммерческим перспективам, направление indoor-навигации становится всё более востребованным и уже привлекло внимание таких крупных игроков на рынке, как Google, Apple, Qualcomm, Broadcom, Sony и другие, и в это, без сомнения, перспективное направление уже инвестируются сотни миллионов долларов [1].

Всемирно известная компания PayPal обозначила свой курс развития, как «инвестиции в мобильные технологии, а также возрождение розничной торговли с помощью, опять же, технологий» [2].

3. Физическая реализация маячка

В плане физической реализации Beacon-маячки – это обычные Bluetooth 4.0 LE (Low Energy) устройства, таким образом, их роль может с успехом выполнять любое устройство, оснащённое BLE-чипом, например, смартфоны на базе Android, а также iPhone, iPad, обычные ноутбуки, Raspberry Pi с usb bluetooth-донглом (электронным ключом) и т.д., на которое установлено специальное приложение, реализующее функции Beacon-маячка [3].

Типичный маячок, показанный на рис. 1, имеет довольно компактные размеры, и способен проработать всего лишь от одной батарейки до двух лет.

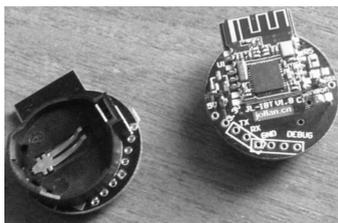


Рис. 1. Маячки, использованные в работе.

4. Формат входных данных

С заданной периодичностью, циклически маячок выдает один и тот же набор данных (рис. 2) [3, 4]:

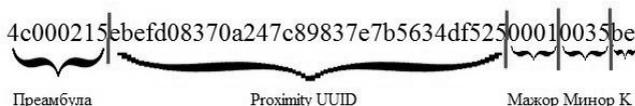


Рис. 2. Пример пакета, передаваемого маячком.

- **Прембула** (4 байта) – префикс пакета, позволяющий установить, что мы имеем дело именно с маячком. Всегда равен 4c000215.

- **Proximity UUID** (16 байт) – Идентификатор группы маяков. Например, если существуют несколько торговых залов, в которых требуется разместить маяки, то во всех этих залах они будут иметь один и тот же UUID, указанный при конфигурации, и это позволит отличать маяки от других, посторонних.

- **Мажор** (2 байта) – позволяет различать небольшой набор маяков внут-ри одной группы. То есть внутри одной большой группы маяков, идентифицируемой UUID, может быть несколько подгрупп, каждая из которых идентифицируется по номеру мажора.

- **Минор** (2 байта) – номер, идентифицирующий сам маяк внутри мажора. Связка uuid+мажор+минор позволяет нам однозначно идентифицировать маяк.

- **TX Power** (параметр K на рисунке выше, 2 байта) – эталонное значение мощности маячка, представляющее собой силу сигнала на расстоянии в 1 метр от маячка. Данная константа используется при определения расстояния от пользователя.

5. Анализ предоставляемого API

Для indoor-навигации Apple предоставляет библиотеку CoreLocation. Она содержит несколько основных сущностей, использованных в работе

- **CLBeacon**. Класс, инкапсулирующий как идентификаторы маяка (uuid, major, minor), так и информацию, используемую при ранжировании (rssi, accuracy, proximity).

- **CLBeaconRegion**. Класс, содержащий идентификаторы группы маяков (uuid, [major, [minor]]). С его помощью можно удобно разделять груп-пы маячков, находящихся, например, на разных этажах, и работать с каждой группой по отдельности.

- **CLLocationManager**. Центральный класс API, который может отслежи-вать маячки в указанном регионе (CLBeaconRegion), рассылая событи-ями как факты входа или выхода в зону действия маячков (monitoring), так и прием пакетов для определения расстояния до источника (ranging).

Таким образом, Apple представила не только формат на BLE-маячки iBeacon, ставший уже широко распространенным, но и удобные средства для работы с ними.

6. Реализация метода, основанного на поиске пересечений сфер

В геометрии трёхмерная проблема трилатерации представляет собой нахождение координат точки пересечения трёх сфер, которые определяются путём решения системы уравнений. Чтобы упростить вычисления, полагаем, что центры всех трех сфер лежат в плоскости $z = 0$, один из них совпадает с началом координат, второй — лежит на оси x . Наложённые ограничения не уменьшают общности: к такому виду может быть приведена любая система соответствующих уравнений путем перехода к другой системе координат. Чтобы найти решение в исходной системе координат, к решению, найденному в этой (приведенной) системе координат, применяются преобразования, обратные к тем, которые позволили исходное множество из трех точек привести в соответствие с ограничениями.

Важно заметить, что задача сводится к задаче о пересечениях окружностей, а не сфер: предпосылкой для этого является закрепление маячков на одной и той же высоте, а также предположение о том, что на одной высоте находятся пользователи. Как результат этих рассуждений, можно отбросить z -координату.

7. Реализация адаптивного геометрического алгоритма

Геометрический алгоритм опирается на взаимоположение окружностей и их радиусы, и результирующие точки пересечения имеют ключевое значение. На рис. 3 показаны наиболее часто возникающие сценарии (точки, соответствующие пересечениям окружностей, обозначены как a , b , c , и так далее).

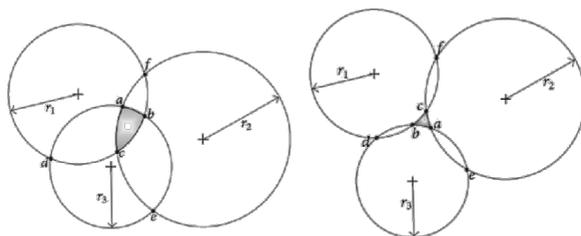


Рис. 3. Наиболее часто встречающиеся сценарии при работе геометрического алгоритма.

В случае, изображенном на рис. 3 (слева), нас интересуют только три точки из шести, так как они формируют область, принадлежащую всем трем окружностям.

На рис. 3 (справа) более сложный сценарий, ведь общей области пересечения всех трех окружностей нет. Следовательно, мы должны воспользоваться другой логикой для нахождения искомой области: для каждой пары точек, образовавшейся в результате пересечения пары окружностей, нас удовлетворяет та точка, которая находится ближе к третьей окружности. Наконец, искомым положением пользователя будет центроид отобранных точек.

В работе [5] представлена идея адаптивного геометрического алгоритма. В сущности, геометрический алгоритм, описанный выше, является частным случаем, а точнее – первой итерацией адаптивного геометрического алгоритма.

После получения набора подходящих точек адаптивная версия геометрического алгоритма пропорционально увеличивает или уменьшает радиусы окружностей для того, чтобы, во-первых, ситуация соответствовала сценарию, приведенному на рис. 3 (слева), и во-вторых, площадь области пересечения была минимальной.

Заключение

В рамках данной работы была разработан готовый программный продукт: библиотека для indoor-локализации для платформы iOS. Согласно выдвинутым требованиям:

- Произведен детальный анализ различных алгоритмов трилатерации, выделены их сильные и слабые стороны;

- В библиотеку включены несколько реализаций алгоритмов, выделенных в результате анализа, а так же предоставлена возможность использовать их комбинации;

- Работа алгоритмов максимально оптимизирована как по скорости работы, так и по объему занимаемой памяти;

- Библиотека легко подключается в iOS-приложение, а ее первоначальная настройка занимает минимальный объем кода.

Однако даже на данном этапе можно выделить направления для дальнейших улучшений. Прежде всего, можно передавать показания акселерометра, магнитометра и гироскопа в качестве управляющих воздействий в фильтр Калмана, и, как результат, во время движения пользователя в большей степени доверять показаниям инерциальной навигационной системы, а во время его остановок – показаниям маяков. Так как навигация внутри помещений может быть трактована как подзадача нахождения положения пользователя в целом, возможна лучшая интеграция с сервисами навигации GPS и «ГЛОНАСС».

Список литературы

1. «Знакомьтесь: биконы!» — Сессия-интенсив про электронные маячки, для бизнесменов и разработчиков [Электронный ресурс]. – Режим доступа : <http://2014.secr.ru/lang/ru/program/submitted-presentations/>
2. «Российские программисты не спали 24 часа» – интервью с Джоном Ланном из paupal [Электронный ресурс]. – Режим доступа : <http://goo.gl/YNqQre>
3. Навигация в помещениях с ibeacon и ИНС [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/post/245325/>
4. Indoor навигация с ibeacon в ios7 [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/company/touchinstinct/blog/195104/>
5. Brida Peter. A novel enhanced positioning trilateration algorithm implemented for medical implant in-body localization [Electronic resource] / Peter Brida, Juraj Machaj // International Journal of Antennas and Propagation. – 2013. – Vol. 2013. – Access mode : <http://dx.doi.org/10.1155/2013/819695>

РЕАЛИЗАЦИЯ МЕТОДИКИ ПРОФИЛАКТИКИ ПРОЦЕССНЫХ РИСКОВ

А.Г. Плотникова

Студент

И.В. Абрамов

Доцент

Введение

Деятельность человека неразрывно связана с рисками, которые в той или иной степени влияют на результат этой деятельности. Основной целью предпринимательской деятельности является достижение стабильного, качественного и запланированного результата. Появление неучтенных источников риска с определённой вероятностью может вызвать потерю предприятием части своих ресурсов, недополучение доходов или появление дополнительных расходов в результате его производственной и финансовой деятельности [1].

В предпринимательской деятельности фундаментальное значение имеет связь риска и прибыли. В качестве основных причин проявления отрицательного воздействия риска является незнание и неосведомленность, случайность и противодействие. Поэтому одной из важнейших задач организационного управления в предприятии является деятельность в рамках эффективного управления рисками, которая состоит

из нескольких этапов: идентификация рисков, обработка определенных рисков (качественная и количественная оценка степени воздействия риска), планирование реагирования на риски, мониторинг и контроль.

Усилия специалистов в данной области были систематизированы и сформулированы в единый международный стандарт [2]. На базе него разработана методика, которая предполагает рассматривать риски по результатам их работы с позиции окружающей среды и с позиции их внутренней идеологии. Автоматизация данной методики поможет добиться ускорения всего процесса управления рисками.

1. Стандарт ГОСТ 3 ИСО/МЭК 31010

Одним из центральных этапов профилактики рисков является их оценка, его основная цель – это представление информации, на основе которой будет принято решение относительно способов обработки риска, реагирования на него и минимизации его отрицательного влияния. Важность реализации методики оценки риска заключается в том, что уже на начальном этапе можно дать оценку степени рискованности деятельности и, приняв определенные управленческие решения, организовать эффективную деятельность организации.

Стандарт ГОСТ 3 ИСО/МЭК 31010 содержит рекомендации по выбору и применению методов оценки риска [2]. В стандарте описаны методы оценки риска, их достоинства, недостатки, приведены указания по применимости методов и рекомендации по выбору методов на основе влияющих факторов. Автоматизация данной методики поможет добиться ускорения всего процесса управления рисками.

2. Анализ методики профилактики процессных рисков (МППР)

Цель рассматриваемой методики – рассмотрение и анализ рисков по результатам их работы с позиции окружающей среды и с позиции их внутренней идеологии.

Цель реализации МППР – это автоматизация процесса управления рисками, позволяющего принять решение о способах и методах обработки рисков и минимизации их отрицательных последствий.

Рассматриваемая в данной статье методика, построена на базе стандарта и использует следующую приведенную в нем информацию [4]:

1. Классификация применимости методов в соответствии с этапами процесса оценки риска – «Директивное использование» (ДИ).

2. Классификация применимости методов в соответствии с факторами, влияющими на выбор метода оценки риска – «Факторное предпочтение» (ФП).

3. Характеристика, отражающая параметры методов и выраженная в констатации положительных/отрицательных сторонах их использования на практике – «Параметрический выбор» (ПВ).

Методика основана на совокупном анализе информации, представленной в ДИ, ФП и ПВ, что позволяет с помощью нее принимать наиболее оптимальные и эффективные решения при выборе методов оценки риска.

На основе анализа достоинств и недостатков всех методов, рассмотренных в стандарте, образованы общие группы, объединившие в себя методы со сходными плюсами и минусами, и выделены общие инвариантные параметры для различных групп (ресурсы, время, достоверность, целесообразность). В итоге образовалось 18 групп преимуществ и 20 групп недостатков с выделенными общими параметрами.

Рассмотрение характеристик параметров методов позволило представить информацию «Параметрический выбор» в виде неполной иерархии [3]. Тогда группы методов представляют собой альтернативы, из которых будет делаться выбор, что позволяет применить при дальнейшем анализе матрицы попарных сравнений, составленных с учетом экспертных оценок. Классификации «Директивное использование» (таблица А.1 [1]) и «Факторы предпочтения» (таблица А.2 [1]) рассматриваются в виде, приведенном в стандарте.

3. Автоматизация процесса, направленного на выбор оптимальных методов оценки риска

Практическая реализация методики заключается в разработке алгоритма выбора методов оценки риска и его реализации программными средствами. Целью создания программы по поиску и выбору оптимального метода является автоматизация процесса принятия решения на основе четко сформулированных классификаций, приведенных в стандарте. Основная идея программы заключается в анализе совокупных критериев, заданных для риска (ДИ, ФП, ПВ) и выбор наиболее оптимального и удовлетворяющего им решения. Данная реализация возможна, так как алгоритм методики построен на использовании математического метода анализа иерархий и логических действий, направленных на выбор из нескольких вариантов в соответствии с заданными критериями.

Разрабатываемое приложение характеризуется следующими свойствами:

1. Входными данными программы являются экспертные оценки групп методов и текущие экспертные оценки критериев по отношению к конкретному рассматриваемому риску. Возможность заполнения этих данных предоставляется пользователю при запуске приложения.

2. Выходными данными являются метод или совокупность методов оценки риска, которые наиболее целесообразно применять в заданных условиях с указанием соответствующих для них критериев (по применимости на каждом этапе оценки – ДИ и степени влияния факторов – ФП), а также с указанием достоинств и недостатков для данного метода.

Выводимая о результате информация позволит лицу, принимающему решение (ЛПР) при необходимости проанализировать работу приложения на правильность и оптимальность результата в заданных условиях.

Входные параметры обрабатываются системой с использованием математического метода анализа иерархий (МАИ). Введенные пользователем данные помещаются в массивы объектов, которые обозначим через $\alpha_1, \alpha_2, \dots, \alpha_n$, каждый массив соответствует определенному критерию (ресурсы, время, достоверность, целесообразность) и один массив содержит оценки самих критериев [5]. Для установления относительной степени предпочтения одного элемента иерархии другому (группы методов) используется шкала отношений, имеющая следующие оценки:

- 1 – объекты равнозначны;
- 3 – преобладание значимости одного объекта над другим;
- 5 – существенная значимость одного объекта над другим.

На основе экспертных оценок групп методов и текущих экспертных оценок критериев по отношению к конкретному рассматриваемому риску с использованием шкалы отношений строятся матрицы парных сравнений. В общем виде матрицу для каждого критерия (ресурсы, время, достоверность, целесообразность) обозначим через $\|\delta_{ij}\|$ следующие коэффициенты –

$$\delta_{ij} = \begin{cases} 1, & \text{если } \dots \alpha_i = \alpha_j, \\ 3, & \text{если } \dots \alpha_i > \alpha_j, \\ 5, & \text{если } \dots \alpha_i \gg \alpha_j, \\ \frac{1}{\delta_{ij}}, & \text{если } \dots \alpha_j > \alpha_i. \end{cases}$$

4. Алгоритм работы программы

Алгоритм работы реализуемой программы можно представить в виде блок-схемы (рисунок).

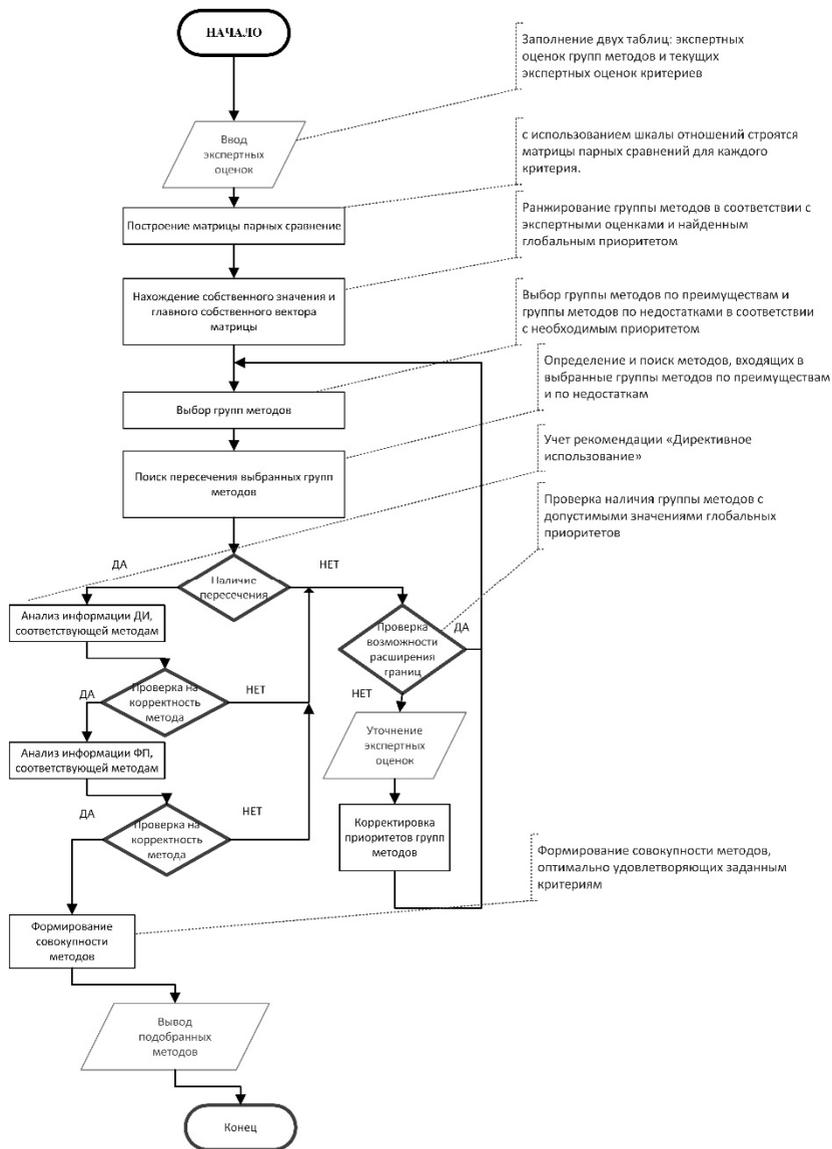


Рисунок. Алгоритм работы программы

Заключение

Таким образом, разрабатываемое приложение, которое является практической реализацией методики выбора методов оценки риска, позволит оптимизировать процесс принятия решения на этапе обработки рисков в системе управления рисками.

Результатом работы приложения является совокупность методов, наиболее полно удовлетворяющих заданным условиям, что позволит проводить эффективную оценку рисков и предоставлять информацию, на основе которой будет принято решение относительно способов обработки риска, его профилактики и минимизации его отрицательного воздействия.

Список литературы

1. Уродовских В.Н. Управление рисками предприятия : Учебное пособие / В. Н. Уродовских. – М. : Вузовский учебник, ИНФРА-М, 2012. – 168 с.
2. ГОСТ Р ИСО/МЭК 31010-2012 Менеджмент риска. Методы оценки риска. – Введ. 2011-12-01. – М. : Стандартиформ, 2012. – 69 с.
3. Саати Т. Принятие решений. Метод анализа иерархий / Т. Саати; пер. с англ. – М. : Радио и связь, 1989. – 316 с.
4. Фирсова О.А. Управление рисками организаций / О.А. Фирсова. – М. : МОО Межрегиональная общественная организация Академия безопасности и выживания, 2014. – 192 с.
5. Шикин Е.В. Математические методы и модели в управлении / Е. В. Шикин, А. Г. Чхартишвили. – М. : Дело, 2000. – 440 с.

АФФИННО-ОДНОРОДНЫЕ ГИПЕРПОВЕРХНОСТИ ПАРАБОЛИЧЕСКОГО ТИПА С ИНДЕФИНИТНОЙ ФОРМОЙ ЛЕВИ

Е.В. Погребецкая

Студент

А.В. Атанов

Ассистент

Введение

Проблема построения классификации голоморфно-однородных, а также аффинно-однородных гиперповерхностей известна с начала XX века (см. [1]). Тем не менее, полное описание в аффинно-однородном случае в пространствах R^3 и C^2 было получено лишь в два последних десятилетия,

а соответствующая задача в C^3 до настоящего момента не решена. В исследованиях, посвящённых данной проблеме, помимо чисто математических методов всё чаще используются алгоритмические и компьютерные подходы к решению указанной задачи.

1. Постановка задачи

Задачей данного исследования является построение фрагментов классификации аффинно-однородных гиперповерхностей $(\varepsilon, \varepsilon)$ -типа в C^3 и изучение эффективности использования базисов Грёбнера для решения данной задачи.

Напомним, что поверхность $M \subset C^3$ называется однородной относительно группы преобразований G , если G транзитивно действует на M . Под аффинной однородностью, соответственно, понимается наличие семейства аффинных преобразований, обеспечивающего свойство транзитивности. Решение поставленной задачи проводится с применением подхода, связанного с анализом канонических уравнений рассматриваемых поверхностей. В работе [2] показано, что уравнение аффинно-однородной строго псевдо-выпуклой поверхности $(\varepsilon, \varepsilon)$ -типа может быть задано в виде

$$v = |z_1|^2 + |z_2|^2 + \varepsilon(z_1^2 + z_2^2 + \bar{z}_1^2 + \bar{z}_2^2) + \sum_{r \geq 3} F_r(z_1, z_2, \bar{z}_1, \bar{z}_2, u), \quad (1)$$

где z_1, z_2, w — координаты в C^3 , $u = \operatorname{Re}(w)$, $v = \operatorname{Im}(w)$, $F_r(z_1, z_2, \bar{z}_1, \bar{z}_2, u)$ — многочлен веса r , где $r = k + l + 2m$ — степень полинома F_r по переменным z_1, z_2 ; l — по \bar{z}_1, \bar{z}_2 , m — по переменной u .

Свойство однородности здесь понимается в локальном смысле, то есть свойство транзитивности должно выполняться лишь в малой окрестности фиксированной точки. В такой ситуации естественным является переход от самих преобразований к их инфинитезимальным (бесконечно малым) аналогам, то есть переход от группы преобразований G к соответствующей алгебре векторных полей, касательных к рассматриваемой поверхности. Каждое такое векторное поле имеет вид $(A_k, B_k, a, b, c, p, s, q$ — комплексные константы)

$$Z = (A_1 z_1 + A_2 z_2 + A_3 w + p) \frac{\partial}{\partial z_1} + (B_1 z_1 + B_2 z_2 + B_3 w + s) \frac{\partial}{\partial z_2} + (a z_1 + b z_2 + c w + q) \frac{\partial}{\partial w}. \quad (2)$$

Помимо вида (2), часто используется и матричное представление векторного поля:

$$Z = \begin{pmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3)$$

Отметим, что в данном случае обсуждаются непрерывные группы преобразований, т. е. группы Ли, с которыми естественно связываются алгебры Ли. Как известно, линейное пространство g матриц, соответствующих линейным полям на поверхности M , образует матричную алгебру Ли, если оно замкнуто относительно операции коммутатора

$$[Z_1, Z_2] = Z_1 \cdot Z_2 - Z_2 \cdot Z_1, \quad (4)$$

где Z_1, Z_2 – матрицы, символ « \cdot » обозначает обычное матричное умножение. Под замкнутостью g относительно коммутатора понимается то, что для любых двух матриц из g их коммутатор также принадлежит g .

Построение классификации аффинно-однородных гиперповерхностей $(\varepsilon, \varepsilon)$ -типа, таким образом, фактически сводится к задаче описания всех комбинаций параметров A_k, B_k, a, b и c (p, s, q в рассматриваемом нами пятимерном случае фиксированы), при которых совокупности матриц вида (3) образуют базис матричных алгебр Ли.

2. Алгоритм построения матричных алгебр Ли

Идея коэффициентного подхода к изучению однородных гиперповерхностей (см. [2]) состоит в получении информации о связях между элементами матриц (3) путем разложения в ряд аналитической функции, стоящей в левой части тождества

$$\operatorname{Re}\{Z(\Phi)\}\big|_M \equiv 0 \quad (5)$$

и последующего анализа весовых компонент в этом разложении. Тождество (5) отражает факт касания полем вида (3) поверхности M .

Алгоритм построения матричных алгебр Ли состоит из следующих шагов.

1. Анализ младших весовых компонент тождества (5) до веса 3 включительно и установление связей между коэффициентами матриц (3).

2. Построение по упрощённому на шаге 1 набору матриц системы нелинейных полиномиальных уравнений.

3. Поиск в данной системе уравнений, разрешимых относительно одной из переменных и их решение.

4. Решение полученной нелинейной системы и получение матричных алгебр Ли, соответствующих однородным гиперповерхностям, задаваемым уравнением (1).

Выполнение первого шага представленного алгоритма позволило получить важную информацию о коэффициентах матриц вида (3). В частности, анализ компонент веса 0 показывает, что константа q всегда вещественна. Рассмотрение слагаемых веса 1 позволяет выразить коэффициенты a и b через p и s :

$$a = 2i(2p\varepsilon + \bar{p}), b = 2i(2s\varepsilon + \bar{s}).$$

Изучение компонент весов 2 и 3 приводит к следующим равенствам:

$$A_1 = f_{11}p + f_{12}s + f_{13}\bar{p} + f_{14}\bar{s} + g_{15}q + \frac{1}{2}\text{Re}(c),$$

$$A_2 = f_{21}p + f_{22}s + f_{23}\bar{p} + f_{24}\bar{s} + g_{25}q - \text{Re}(A_2),$$

$$A_3 = g_{11}p + g_{12}s + g_{13}\bar{p} + g_{14}\bar{s} + b_{15}q,$$

$$B_1 = f_{31}p + f_{32}s + f_{33}\bar{p} + f_{34}\bar{s} + g_{35}q - \text{Re}(A_2),$$

$$B_2 = f_{41}p + f_{42}s + f_{43}\bar{p} + f_{44}\bar{s} + g_{45}q + \frac{1}{2}\text{Re}(c),$$

$$B_3 = g_{21}p + g_{22}s + g_{23}\bar{p} + g_{24}\bar{s} + b_{25}q.$$

Здесь f_{ij}, g_{ij}, b_{ij} — комплексные коэффициенты, связанные с полиномами F_3, F_4, F_5 уравнения (1) соответственно.

После упрощения матричных алгебр осуществляется переход ко второму шагу. Условие замкнутости (4) сводит задачу поиска алгебр Ли к решению большой системы нелинейных полиномиальных уравнений, решения которой дают значения параметров, при которых наборы матриц вида (3) образуют алгебру Ли.

В рассматриваемом случае была получена система из 133 вещественных полиномиальных уравнений относительно 30 вещественных неизвестных. Часть из этих уравнений можно разрешить и выразить достаточно большое количество неизвестных, которых в итоге остаётся 16. Тем не менее, решение полученной системы представляет собой очень сложную задачу. Как известно, подходов к решению систем полиномиальных уравнений общего вида не существует. В связи с этим полное решение рассматриваемой задачи пока не получено, однако и изучение частных случаев представляет достаточно большой интерес. В работе рассматривается один из таких частных случаев, в котором 4 из 16 неизвестных полагаются равными нулю. Даже в этом случае система нелинейных уравнений остаётся крайне громоздкой и возникает потребность в её упрощении. Для этого в данной работе привлекаются

базисы Грёбнера (см. [3]), эффективность использования которых в решении поставленной задачи не исследовалась.

3. Применение базисов Грёбнера для решения систем полиномиальных уравнений

Как было отмечено выше, решение систем нелинейных полиномиальных уравнений представляет собой нетривиальную задачу. Для облегчения её решения в последние годы часто используются базисы Грёбнера.

Определение. Непустое подмножество I кольца многочленов $K[x_1, x_2, \dots, x_n]$ называется идеалом в K , если для любых элементов $p, q \in I$ элемент $(p - q) \in I$, а также для любых $p \in I, c \in K$ элемент $p \cdot c \in I$.

Определение. Множество полиномов g_i является базисом Грёбнера для полиномиального идеала I , если старший член любого из полиномов идеала I делится на один из старших членов полиномов g_i .

Как известно (см. [3]), множество решений системы уравнений, полученной из базисов Грёбнера, эквивалентно множеству решений системы, для которой этот базис был построен. В некоторых случаях использование базисов Грёбнера существенно облегчает поиск решения системы. Однако даже самые быстрые алгоритмы, строящие базис Грёбнера, на больших системах работают крайне медленно и зачастую базис не может быть вычислен за приемлемое время.

В связи с этим при компьютерной реализации рассматриваемого алгоритма в системе Maple базисы Грёбнера строились не для всей системы, а лишь для некоторого небольшого числа её уравнений. Перебирались возможные комбинации из двух, трёх, четырёх и иногда большего числа уравнений, при этом с помощью написанной программы оценивалось время вычислений, и если оно превышало заданный порог, осуществлялся переход к следующей комбинации. Кроме того, в программе сделана проверка полученных в базисе Грёбнера полиномов на их разложимость на множители. Оказалось, что во многих случаях базис Грёбнера имеет гораздо более простой вид, чем исходная система.

Например, подсистема, состоящая из уравнений

$$\begin{aligned} \varepsilon(2t_2t_5 + t_2t_9 - 4t_4t_5 - 2t_4t_9) - t_1t_2 - t_2t_5 - t_4t_5 &= 0, \\ \varepsilon(2t_2t_5 + t_2t_9 - 4t_4t_5 - 2t_4t_9) - t_1t_6 - 3t_2t_5 - t_2t_9 + 3t_4t_5 + 2t_4t_9 &= 0, \\ \varepsilon(2t_2t_5 + t_2t_9 - 4t_4t_5 - 2t_4t_9) - t_1t_6 - t_2t_5 + 5t_4t_5 + 3t_4t_9 &= 0, \\ \varepsilon(2t_2t_5 + t_2t_9 - 4t_4t_5 - 2t_4t_9) - t_1t_6 - t_2t_5 - t_4t_5 &= 0, \\ 2\varepsilon^2(2t_2t_5 + t_2t_9 - 4t_4t_5 - 2t_4t_9) - \varepsilon(2t_1t_6 + 4t_2t_5 + t_2t_9 - 14t_4t_5 - 8t_4t_9) &+ \end{aligned}$$

$$+t_1t_6 + t_2t_5 - t_4t_5 - 2t_4t_7 = 0,$$

порождает базис Грёбнера вида

$$\begin{aligned} t_4(2t_7 - t_9) &= 0, \\ 2t_1t_6 - t_2t_9 - t_4t_9 &= 0, \\ t_4(2t_5 + t_9) &= 0, \\ t_2(2t_5 + t_9) &= 0. \end{aligned}$$

Очевидно, что решение второй системы гораздо проще, чем первой. Однако для решения всей исходной системы полиномиальных уравнений требуется рассматривать очень большое число подслучаев, не во всех из которых базис Грёбнера имеет такой простой вид.

4. Классификационные результаты

Используя описанный подход, были построены примеры пятимерных матричных алгебр Ли, соответствующих однородным поверхностям $(\varepsilon, \varepsilon)$ -типа. Приведём пример одной из полученных алгебр (здесь $\mu = (2\varepsilon + 1)$, $\omega = (2\varepsilon - 1)$, $a \in R$):

$$\begin{aligned} E_1 &= \begin{pmatrix} 0 & -\frac{ia\mu}{\varepsilon} & -\frac{ia^2\mu}{\varepsilon^2} & 1 \\ -\frac{ia\mu}{\varepsilon} & 0 & 0 & 0 \\ 2i\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & \frac{a\mu}{\varepsilon} & \frac{a^2\mu}{2\varepsilon^2} & i \\ \frac{a\omega}{\varepsilon} & 0 & 0 & 0 \\ -2\omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_3 &= \begin{pmatrix} -\frac{ia\mu}{\varepsilon} & 0 & 0 & 0 \\ 0 & -\frac{2ia\mu}{\varepsilon} & -\frac{ia^2\mu}{2\varepsilon^2} & 1 \\ 0 & 2i\mu & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} 2a & 0 & 0 & 0 \\ 0 & 4a & \frac{a^2\mu}{2\varepsilon^2} & i \\ 0 & -2\omega & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_5 &= \begin{pmatrix} -\frac{ia^2\mu}{2\varepsilon^2} & 0 & 0 & 0 \\ 0 & -\frac{ia^2\mu}{2\varepsilon^2} & 0 & 0 \\ 0 & 0 & -\frac{ia^2\mu}{2\varepsilon^2} & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

Заключение

В работе было установлено, что использование базисов Грёбнера позволяет в значительной степени упростить решение поставленной задачи в некоторых случаях. Используя описанный подход, были построены примеры пятимерных матричных алгебр Ли, соответствующих однородным поверхностям $(\varepsilon, \varepsilon)$ -типа в S^3 . Для работы была привлечена система символьной математики Maple, в которой были реализованы алгоритмы, соответствующие описанным выше этапам.

Список литературы

1. Cartan E. Sur la géométrie pseudoconforme des hypersurfaces de deux variables complexes / E. Cartan // Ann. Math. Pura Appl. – 1932. – №4 (11). – P. 17-90.
2. Лобода А.В. Аффинно-однородные вещественные гиперповерхности 3-мерного комплексного пространства / А.В. Лобода // Вестник Воронежского государственного университета. Серия : Физика. Математика. – Воронеж, 2009. – № 2. – С. 71-91.
3. Аржанцев И.В. Базисы Гребнера и системы алгебраических уравнений / И.В. Аржанцев. – М. : МЦНМО, 2003. – 68 с.

РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ ПОЛУЧЕНИЯ ДАННЫХ ИЗ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»

М. В. Полев
Студент
А. В. Сычев
Доцент

Введение

Рост популярности делает социальные сети с каждым годом все более и более привлекательным для исследователей местом [1]. С увеличением количества пользователей увеличивается объем данных, доступных для анализа. Встает вопрос о способе получения этих данных. Эту проблему можно рассмотреть с нескольких сторон:

1. Поиск массивов данных, выкладываемых другими исследователями;
2. Попытка получить данные, взаимодействуя с компаниями, специализирующимися на анализе данных подобного рода;

3. Использование имеющихся коммерческих решений, специализирующихся на сборе данных из социальных сетей;
4. Парсинг HTML страниц профилей социальных сетей;
5. Использование интерфейса программирования приложений (API), предоставляемого социальными сетями.

Основной недостаток первого способа – ограниченность получаемых данных. Исследователю придется пользоваться только теми данными, что удастся найти. Вместе с тем позволяет получить большой объем готовых данных с большой скоростью.

Во втором способе зачастую не заинтересованы сами компании, а процесс получения хотя бы первоначальных данных сильно затягивается.

Третий способ не подходит для начальных этапов исследований, когда бюджет практически отсутствует, и выбираемые данные строго ограничены алгоритмом выборки данных, закодированным в этих программных средствах.

Четвертый способ требует хорошего знания структуры html документа, сложен в реализации и сбор данных занимает много времени. Плюс ко всему возможно получить только те данные, что отображены на странице в профиле. Добавление каждой новой страницы для парсинга уменьшает скорость выборки данных.

Пятый способ дает наиболее гибкий инструмент для сбора данных, но требует знаний какого-либо языка программирования. К тому же скорость получения данных сильно ограничена, так как существуют ограничения на количество запросов к серверу социальной сети за единицу времени.

1. Постановка задачи

В последние годы происходит активное привлечение инструментов Social Mining для решения бизнес-задач. Согласно исследованию [2] ежемесячно в социальных сетях в России появляется 620 млн сообщений от 35 млн авторов. Многие пользователи открыто публикуют информацию о себе, своих интересах и предпочтениях, поэтому социальные сети являются для коммерческих компаний мощным инструментом маркетинговых исследований и акций (например, для таргетированной рекламы своих продуктов и услуг клиентам, в том числе потенциальным).

При всей популярности исследования социальных сетей за рубежом, на данный момент не так много российских исследователей заняты проблемой анализа социальных сетей. Это приводит к тому, что не существует готовых массивов данных доступных для скачивания и анализа, а реализованные коммерческие продукты не удовлетворяют требованию к объему выбираемых данных. Не удовлетворившись ни одним из найденных для сбора данных средством, автор статьи принял

решение о необходимости реализации собственного программного продукта.

2. Архитектура программного решения

Для формирования коллекции данных был использован программный интерфейс Open API (<https://new.vk.com/dev/openapi>), предоставляемый сетевой платформой ВКонтакте для выполнения запросов на получение информации о пользователях и группах ВКонтакте. Использование интерфейса программирования приложений подразумевает реализацию программного средства, которое будет взаимодействовать с сервером социальной сети. Подход является очень гибким, так как существует возможность закодировать получение именно тех данных, которые необходимы для текущего исследования, в необходимом объеме.

Программное приложение было реализовано на языке C#. Данные, полученные из социальной сети, сохраняются в СУБД PostgreSQL. Для удобства использования интерфейса программирования приложений была использована обертка, реализованная в библиотеке VkNet (<https://vknet.github.io/vk/>).

Система была реализована в виде многопоточного консольного приложения (работающего в операционной системы Windows). Общая схема реализованной системы представлена на рис. 1.

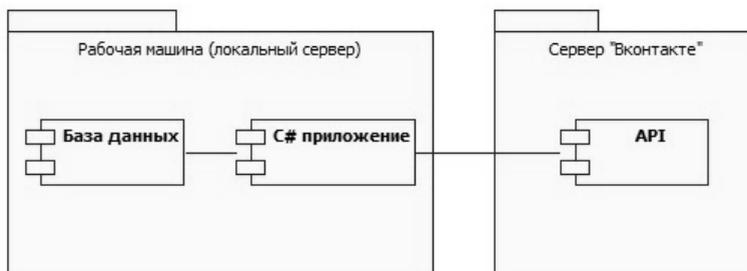


Рис. 1. Схема компонентов системы

Из схемы видно, что консольное приложение является связующим звеном между сервером с данными, и локальной базой данных. Для взаимодействия с сервером социальной сети используется интерфейс программирования приложений (API) предоставляемый социальной сетью.

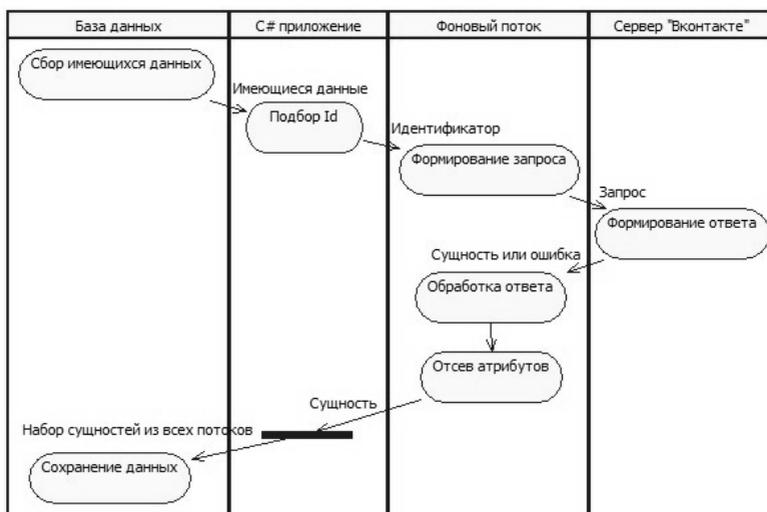


Рис. 2. Схема работы системы считывания данных

Общая схема работы системы считывания данных представлена на рис. 2. Алгоритм начинает работу со считывания из базы данных значений идентификаторов уже имеющихся сущностей. После этого осуществляется подбор случайного значения, которому потенциально будет соответствовать определенная сущность на сервере социальной сети. После подбора случайного значения, открывается новый поток, в котором идет обращение к серверу социальной сети «Вконтакте».

Получение сущности из сервера социальной сети осуществляется посредством HTTP запроса на сервер социальной сети. Интерфейс программирования приложений используется для взаимодействия сервера социальной сети со сторонними приложениями. Тело запроса должно содержать служебные данные, которые могут подтвердить факт прохождения аутентификации приложением, отправляющим запрос, тип выбираемой сущности (в случае текущего исследования – сообщество или пользователь), а также идентификатор сущности, которому предположительно соответствует сущность базы данных сервера социальной сети.

После получения ответа сервера поток передает управление основному потоку приложения, который осуществляет запись сущности в базу данных, или обработку ошибки. Процесс повторяется в бесконечном цикле до тех пор, пока не будет найдено ни одно случайное значение,

отсутствующее в локальной базе данных, и соответствующее реальной сущности сервера социальной сети.

Однако очевидно, что у данного подхода также имеются недостатки. Главный из них – относительно малая скорость выборки данных. Проблема возникает из-за ограничений интерфейса программирования приложений, который позволяет осуществлять не более трех запросов в секунду, причем объем выбираемых за запрос данных строго ограничен.

Механизм многопоточности позволяет частично преодолеть проблему ограничения скорости выборки данных, так как параллельное выполнение множества потоков позволяет получать в секунду гораздо большее количество данных.

3. Тестирование

Помимо реализации программного решения по получению данных социальной сети, было проведено небольшое исследование собранной из социальной сети ВКонтакте коллекции данных, с целью тестирования работоспособности реализованного приложения.

Для проведения предварительного исследования были выбраны атрибуты профилей, имеющие числовой тип данных. Профили пользователей содержали следующие атрибуты:

- A_i – количество альбомов, которые содержатся на страницах каждого конкретного пользователя.

- M_i – количество аудиозаписей, которые размещены в профилях каждого конкретного пользователя.

- Ph – количество фотографий, которые размещены в профилях каждого конкретного пользователя.

- V_i – количество видеозаписей, которые размещены в профилях каждого конкретного пользователя.

- F_o – количество других пользователей, которые подписаны на обновления данного пользователя.

- S_u – количество профилей, на которые подписан каждый конкретный пользователь.

- N_o – количество заметок (текстовых записей в свободной форме, составленных пользователем).

- P_a – количество страниц, на которые подписан каждый конкретный пользователь.

- F_r – количество друзей данного пользователя.

- G_i – количество подарков от других пользователей.

- G_r – количество сообществ, в которых состоит каждый конкретный пользователь.

– PWU – количество фотографий, на которых отмечен данный пользователь.

– VWU – количество видеозаписей, на которых отмечен данный пользователь.

В таблице 1 приведены значения средних и медианы для числовых атрибутов профилей пользователей.

Таблица 1

Характеристики профилей пользователей социальной сети

	Доступны	Имеют	Среднее значение	Медиана	
Al	1736141	417521	0.805	0	
Mu	1736141	480561	31.272	0	
Ph	1736141	1064065	34.520	1	
Vi	1736141	586298	35.294	0	
Fo	1736141	1032488	29.460	1	
Su	1736141	488132	5.737	0	
No	1736141	72435	0.328	0	
Pa	1736141	757681	17.335	0	
Fr	1736141	1015753	47.245	2	
Gi	1445932	229700	1.742	0	
Gr	1346670	594554	18.873	0	
PWU	1343209	88677	0.912	0	
VWU	1445093	56030	0.489	0	
Всего пользователей считано:				2334020	100%
Из них удалено:				597879	25,6%
Все атрибуты доступны у:				1278827	54,8%

Столбец «Доступны» содержит информацию о количестве сущностей, которые содержат атрибут в свободном доступе, то есть значение атрибута не скрыто настройками приватности. Столбец «Имеют» содержит информацию о количестве сущностей, для которых атрибут является «Доступным», и которые имеют более нуля значений этого атрибута.

Аналогично, для апробации взаимодействия приложения с сервером социальной сети с целью получения информации о профилях сообществ, было считано некоторое количество данных сообществ «Вконтакте». Для профилей данных этого типа были собраны значения следующих атрибутов:

- MC – количество подписчиков данного сообщества.
- Al – количество альбомов данного сообщества.
- Ph – количество фотографий данного сообщества.
- Vi – количество видеозаписей данного сообщества.

- Mu – количество аудиозаписей данного сообщества.
- To – количество обсуждений данного сообщества.
- Do – количество документов данного сообщества.

Результаты расчета по профилям сообществ социальной сети «ВКонтакте» представлены в таблице 2.

Таблица 2

Характеристики атрибутов профилей сообществ социальной сети

	Доступны	Имеют	Среднее значение	Медиана	
MC	1795314	1366811	122.871	2	
Al	1178864	1178577	1.840	1	
Ph	1178864	691591	54.416	1	
Vi	562628	227242	7.123	0	
Mu	500459	244913	8.035	0	
To	562303	266161	1.795	0	
Do	171385	21932	0.631	0	
Всего сообществ считано:				2038769	100%
Все атрибуты доступны у:				132729	6,5%
Заблокировано:				56246	2,6%
Удалено:				49555	2,4%
Публичных:				1656435	81,3%
Закрытых:				316277	15,5%
Приватных:				66057	3,3%
Событий:				321579	15,8%
Групп:				1430802	70,2%
Страниц:				286388	14,1%

Заключение

В статье была представлена общая схема работы приложения, осуществляющего выборку данных из социальной сети «ВКонтакте». Показано, как механизм многопоточности позволяет частично преодолеть ограничения на количество запросов за единицу времени, накладываемое настройками сервера социальной сети. Также представлены данные, которые удалось получить с использованием такого подхода, и пример небольшого анализа, проведенного на основании этих данных.

Поскольку принципы хранения данных крупных интернет ресурсов в большинстве своем основаны на использовании реляционных баз данных, подобная схема может быть применена и к другим, как российским, так и зарубежным социальным сетям.

Таким образом, можно сказать, что сфера анализа социальных сетей не является доступной лишь для крупных компаний, занимающихся

проблемами анализа большого объема данных. Небольшие навыки программирования, а также предоставляемые социальными сетями интерфейсы программирования приложений, позволяют любому желающему осуществить сбор и анализ интересующих его данных, такой, как, например, в [3].

Список литературы

1. Давыдов А.А. Системная социология: Social Networks Mining [Электронный ресурс] / А.А. Давыдов. – Электрон. текстовые дан. – М. : ИС РАН, 2009. – Режим доступа : http://www.isras.ru/index.php?page_id=1033
2. Brand Analytics : Российская аудитория социальных сетей весной 2015 года [Электронный ресурс]. – Режим доступа : <http://adindex.ru/news/researches/2015/05/25/123830.phtml>
3. Сычев А.В. Сравнительное изучение атрибутов профилей сообществ LiveJournal в 2010 и 2011 годах // Электронные библиотеки : перспективные методы и технологии, электронные коллекции : тр. 13-ой Всерос. науч. конф. RCDL 2011, Воронеж, 19-22 окт. 2011 г. – Воронеж, 2011. – С. 44-52. (<http://ceur-ws.org/Vol-803/paper7.pdf>)

РАЗРАБОТКА МОДУЛЕЙ АНАЛИЗА ДАННЫХ И ОТМЕНЫ ДЕЙСТВИЙ ДЛЯ ПРОГРАММНОГО КОМПЛЕКСА DYNAMICS

А.С. Пономарев

Студент

П.С. Лысачев

Старший преподаватель

Введение

Развитие современной энергетики сопровождается повышением требований к качеству испытаний энергетических установок. Специалистам требуются новые инструменты для повышения эффективности анализа результатов. С этой целью сотрудниками Атомтехэнерго был разработан программный комплекс Dynamics. Он позволяет проводить эффективный анализ результатов испытаний и производить специальные расчеты.

С целью повышения эффективности работы специалистов в программный комплекс Dynamics был внедрен модуль отмены действий и компонент корреляционного анализа с модулем подготовки данных.

1. Dynamics 4.1

Целью данной работы разработка и внедрение компонентов анализа данных и модернизации интерфейса программного комплекса Dynamics.

Программное обеспечение Dynamics 4.1 специально разработано для быстрого построения сложных графиков и эффективного анализа поведения параметров во время динамических испытаний на АЭС. Кроме того, в программном обеспечении заложены функции для расчета нейтронно-физических характеристик активной зоны реактора при работе на минимально-контролируемом уровне мощности, такие как:

- Расчет симметричности загрузки,
- Расчет дифференциальной эффективности,
- Расчет температурного/барометрического коэффициента реактивности,
- Расчет мощностного эффекта,
- Проверка сцепленности.

2. Модернизация интерфейса программного комплекса Dynamics

Интерфейс пользователя — это разновидность интерфейса, в котором одна из сторон представлена человеком, а другая машиной [1]. Вопреки всеобщему предубеждению, пользовательский интерфейс выражен не только графической частью или оформлением рабочей зоны. Он включает в себя также поведение системы, основные элементы взаимодействия с пользователем, прогнозирование действий пользователя.

Каждое действие пользователя или машины должно быть обработано и не нести опасности для пользовательских данных. Удаление информации должно происходить лишь по желанию пользователя с возможностью отмены изменений. Внедрение подобного функционала необходимо произвести с минимальными вмешательствами в существующий исходный код программы. Ввиду сложности реализации алгоритма, при помощи сохранения всего изменяемого объекта, было принято решение разработать процедуру отмены для каждого отдельного действия.

Модуль включает два списка действий. UndoList содержит все выполненные пользователем действия. RedoList содержит отмененные действия. При каждом действии создается экземпляр класса, соответствующий выполняемому действию. Объект содержит измененные (старые) данные. В зависимости от класса объекта наборы этих данных могут отличаться.

Каждое действие в модуле представлено отдельным классом, имеющим общий родительский класс TAct.

Каждый класс имеет поля для сохранения изменяемых параметров и содержит основные методы:

- DoAct – выполняет действие с сохраненными параметрами. Необходим для действий, у которых отмена аналогична действию и отличается лишь параметрами.
- Undo – отмена действия.
- Redo – повтор действия (отмена отмены).
- Action – сохранение изменяемых параметров.

3. Создание компонента для анализа зависимостей между параметрами

Компонент DependenceParamFromParam предназначен для оценки зависимостей между исследуемыми параметрами. После анализа предметной области, выбора исследуемых переменных и выбора типа исследования необходимо произвести подготовку данных. Подготовка данных включает в себя очистку данных и приведение их к требуемому виду. Ошибка на этапе подготовки может вызвать существенные отклонения от действительности модели или ошибки при формировании выводов из анализа. Модуль подготовки данных включает в себя следующие функции:

- Подготовка массивов для построения геометрического представления;
- Аппроксимация рядов;
- Линейная интерполяция;
- Аппроксимирующий полином Ньютона для неравных промежутков;
- Поиск ближайшего массива;
- Скользящая средняя;
- Сглаживание функций;
- Поиск вершин;
- Приведение ряда с неравными промежутками к нормальному виду.

Из известных интерполяционных формул для решения поставленных задач подходит только интерполяционный многочлен Ньютона, так как остальные рассчитаны на равномерную или специальную дискретизацию [2].

Компонент DependenceParamFromParam использует подготовленные массивы данных. Визуализация происходит на плоскости с возможностью изменения временных промежутков, типов отображения и способа аппроксимации.

Результатом работы модуля является линейная модель зависимости параметров и коэффициент корреляции.

Модуль подготовки данных может использоваться в других частях программы. Задача приведения временных рядов к равномерно дискретизированным позволяет расширить возможности анализа данных. Приведение ряда с неравномерными временными промежуткам в подобный вид может сопровождаться потерями значений и появлением погрешности. В связи с этим необходимо провести анализ погрешности.

Выбор метода приведения ряда зависит от типа данных. В случае если значения переменных принадлежат множеству действительных чисел, применяются следующие методы.

Аппроксимирующий полином Ньютона. Строится для нескольких точек, путем нахождения разделенной разности Δ для каждой пары последовательных значений

$$\Delta_{1n} = \frac{e(t_n) - e(t_{n-1})}{t_n - t_{n-1}},$$

где Δ_{1n} – разделенная разность первого порядка, n – номер элемента, t_n – время в точке n , $e(t_n)$ – значение переменной в момент времени t_n .

После этого находят разделенные разности высших порядков.

$$\Delta_{n1} = \frac{\Delta_{(n-1)2} - \Delta_{(n-1)1}}{t_n - t_0}.$$

Интерполяционный многочлен Ньютона для неравных промежутков будет определять алгебраическим многочленом n -й степени [2]:

$$P_n(t) = e(t_0) + (t - t_0)\Delta_{11} + \dots + (t - t_0)\dots(t - t_n)\Delta_{n1}.$$

Второй не менее эффективный метод – метод линейной интерполяции. Суть данного метода заключается в поиске промежуточных значений на прямой, проведенной через две точки.

Выбор метода должен осуществляться пользователем в зависимости от типа данных.

Если исследуемая переменная принадлежит ограниченному множеству дискретных значений, то следует использовать округление в сторону значения с меньшим отклонением.

После получения временного ряда с равными временными промежутками необходимо оценить погрешность. Оценивая результаты, пользователь может изменять метод интерполяции либо степень аппроксимирующего полинома, с целью поиска наибольшей точности.

Также можно произвести аппроксимировать данные, предварительно проведя сглаживание. Это позволит добавить на график линию тренда. После этого на графике выбираются точки, по которым будет проводиться

аппроксимация. Автоматический выбор точек может быть обусловлен положением экстремумов либо заранее определенными временными промежутками. Также имеется возможность обозначить эти точки вручную. В случае, когда временные промежутки между точками равны, можно произвести аппроксимацию методом наименьших квадратов. Если промежутки разные, то рекомендуется использовать интерполяционный многочлен Ньютона. Все предложенные методы сопровождаются визуализацией и оценкой погрешности.

Для корреляционного анализа приведение рядов к равномерной дискретизации производится парами параметров, где в качестве искомым промежуточных значений выступают значения времени парного параметра. В результате получается набор пар переменных для каждого момента времени. Полученные данные можно представить в графической форме, как показано на рисунке.

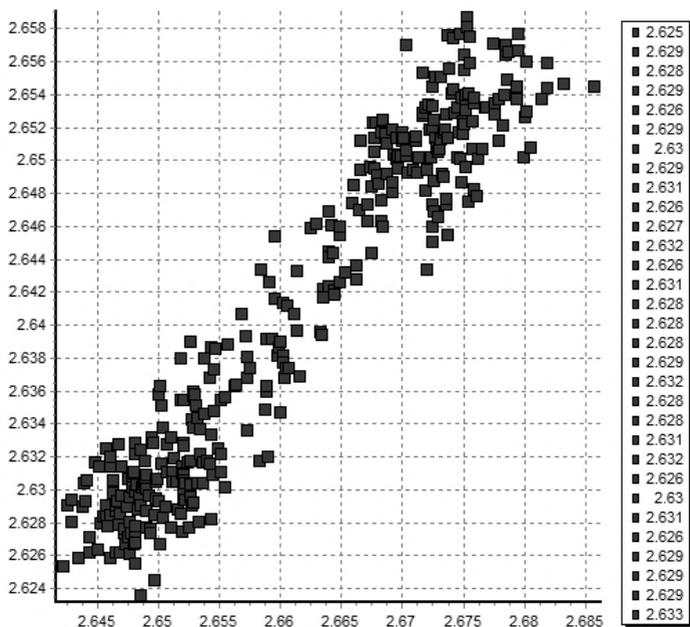


Рисунок. Графическое представление зависимости параметров

Результатом работы является расчет коэффициента корреляции. Расчет производится по формуле Пирсона:

$$r = \frac{\sum_{i=0}^n ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=0}^n ((x_i - \bar{x})^2 (y_i - \bar{y})^2)}}$$

где x_i, y_i – текущие значения в ряду, \bar{x}, \bar{y} – значения математического ожидания, n – количество переменных в ряду.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i,$$

К коэффициенту корреляции Пирсона следует относиться с осторожностью, поскольку аномальные значения исследуемых переменных могут существенно исказить результат. Избежать этого можно путем визуального анализа зависимости и исключения из анализа аномальных значений.

Заключение

Данная статья посвящена разработке компонентов анализа данных и модернизации интерфейса программы Dynamics. В интерфейс программы были внесены изменения, которые гарантированно повысят эффективность труда специалистов. Также в функционал программы был добавлен модуль построения зависимости между параметрами. В ходе работы был создан универсальный модуль подготовки данных, который будет использован при дальнейшей разработке компонентов, что, при условии постоянного развития энергетической отрасли, является неизбежным.

Список литературы

1. Лекционный материал ГКПОУ [Электронный ресурс] : многопредмет. науч. журн. / Прокопьевский. гор.-техн. тех. – Электрон. журн. – Режим доступа : <http://www.pgtk.edu.ru>
2. Спектральный анализ цифровых сигналов с неравномерной дискретизацией / В. Н. Булатов [и др.] // Вестник ОГУН. – 2006. – №6. – С. 185-190.

МОДЕЛИ И АЛГОРИТМЫ СУПЕРПИКСЕЛЬНОЙ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

С.В. Саввин
Студент
А.А. Сирота
Профессор

Введение

Одной из наиболее часто решаемых задач при предварительной обработке изображения является его сегментация – разбиение изображения на непересекающиеся области, покрывающие все изображение и однородные по некоторым признакам: цвет, яркость, текстура и т. д.

Многие алгоритмы обработки изображений используют представление изображения в виде пиксельной сетки. Однако это не всегда оптимально и обусловлено в большей степени хранением изображения в цифровой форме.

Более естественным является разбиение изображения на множество мелких частей (суперпикселей), представляющих из себя группы расположенных рядом и схожих между собой пикселей. Суперпиксель является атомарным регионом изображения, т. е. все входящие в него пиксели рассматриваются при дальнейшей обработке, как единое целое. При этом суперпиксели не обязательно должны иметь строго определенную форму (рис. 1).



Рис. 1. Пример суперпиксельной сегментации изображения

1. Преимущества суперпиксельной сегментации

Суперпиксельная карта изображения, полученная после проведения суперпиксельной сегментации, обладает рядом преимуществ по сравнению с обычной сеткой пикселей [1].

Во-первых, это вычислительная эффективность: десятки и сотни тысяч пикселей заменяются всего лишь на десятки или сотни суперпикселей, которые выступают в алгоритмах как единое целое, что может значительно снизить время обработки в алгоритмах распознавания.

Во-вторых, в суперпиксельном представлении изображения можно говорить о взаимосвязях между отдалёнными друг от друга пикселями, в то время как в пиксельном – только о связях между рядом лежащими пикселями.

В-третьих, суперпиксели имеют большее смысловое значение. Каждый суперпиксель является согласованной единицей, так как принадлежащие ему пиксели имеют схожие цвета, яркость и прочие свойства.

И наконец, в отличие от окна прямоугольной формы, суперпиксель с большей вероятностью принадлежит только одному объекту изображения, не затрагивая остальные.

Данные свойства суперпикселей обуславливают их использование при решении задач сегментации объектов как с известными, так и с неизвестными свойствами.

2. Алгоритмы суперпиксельной сегментации

Основной подход к сегментации – пиксели внутри каждого суперпикселя должны быть максимально похожи, а пиксели, находящиеся в разных суперпикселях могут сильно отличаться. Данная задача может решаться принципиально разными способами.

Все алгоритмы суперпиксельной сегментации можно разделить на три большие группы [2]:

- Разделяющие;
- Объединяющие (агломеративные);
- Дискриминационные.

Разделяющие и объединяющие алгоритмы суперпиксельной сегментации представляют изображение в виде ненаправленного взвешенного графа $G = (V, E)$, где вершинами V служат пиксели, а грани E соединяют смежные пиксели. Весом грани $w(i, j)$ служит мера сходства вершин i и j , которые она соединяет.

В разделяющих алгоритмах всё изображение изначально считается одним большим суперпикселем, а затем осуществляется его рекурсивное разбиение на более мелкие компоненты.

К разделяющим алгоритмам относится, например, первый опубликованный алгоритм суперпиксельной сегментации – алгоритм нормализованного разреза (Ncut) [3]. Его идея заключается в поиске на графе минимального нормализованного разреза между двумя половинами A и B графа $G = (V, E)$ (1). Под разрезом понимается суммарный вес всех граней, соединяющих две половины графа (2). Учёт размера кластера осуществляется через сумму весов всех его граней (3).

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}, \quad (1)$$

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} \omega(u, v), \quad (2)$$

$$\text{assoc}(A, V) = \sum_{u \in A, t \in V} \omega(u, t). \quad (3)$$

Объединяющие алгоритмы, напротив, принимают каждый пиксель изображения за отдельный суперпиксель, осуществляя в дальнейшем их жадное слияние.

Примером объединяющего алгоритма является алгоритм ERS (Entropy Rate Superpixel) [4], в котором осуществляется максимизация целевой функции $F(A)$ (4).

$$F(A) = B(A) + \lambda \cdot H(A), \quad (4)$$

где $A \subset E$ – подмножество рёбер, образующих разбиение на суперпиксели.

Целевая функция состоит из двух основных компонентов, вес которых настраивается при помощи коэффициента λ :

- $H(A)$ – энтропия случайного пути на графе $G = (V, A)$, отвечающая за однородность пикселей внутри суперпикселя;
- $B(A)$ – функция балансировки, отвечающая за получение суперпикселей примерно одинакового размера.

В дискриминационных алгоритмах осуществляется предварительное разбиение пикселей на кластеры с последующим итеративным уточнением.

Так, алгоритм SLIC (Simple Linear Iterative Clustering) [5] представляет из себя модифицированный алгоритм кластеризации k -средних, т.е. в нём осуществляется минимизация функции ошибки кластеризации. Основное отличие SLIC от k -средних заключается в ограничении области поиска: пиксели для каждого сегмента ищутся не во всём изображении, а в небольшой области, пропорциональной размеру сегмента.

Таким образом, задача суперпиксельной сегментации сводится к задаче оптимизации значения той или иной целевой функции. При этом

результат суперпиксельной сегментации может существенно отличаться в зависимости от используемого подхода. Выбор того или иного алгоритма зависит от вида решаемой задачи и, соответственно, выдвигаемых к нему требований.

3. Критерии оценки качества

Так как существует множество алгоритмов, отличающихся по получаемым результатам, крайне важной является задача сравнения различных алгоритмов.

Не все критерии оценки качества обычной сегментации подходят для оценки суперпиксельной сегментации. Поэтому часть из них заменяется на новые, а также добавляются новые критерии (например, для оценки компактности).

Можно выделить следующие критерии для оценки качества суперпиксельной сегментации:

- Время работы алгоритма (чем меньше, тем лучше);
- Точность:
 - Процент обнаруженных границ (относительное количество реальных границ объектов, находящихся рядом с границами суперпикселей; чем больше, тем лучше);
 - Избыточность сегментации (ошибка, связанная с прохождением реальных границ объектов внутри суперпикселей; чем меньше, тем лучше);
 - Максимально достижимая точность сегментации (точность обычной сегментации, выполненной на основе суперпиксельной; чем больше, тем лучше);
- Компактность:
 - Среднеквадратичное отклонение размеров суперпикселей (чем меньше, тем лучше);
 - Изодиаметрический и изопериметрический коэффициенты (отношение площади суперпикселя к площади круга, имеющего такой же диаметр и периметр соответственно; чем больше, тем лучше);
- Устойчивость к преобразованиям и шумам (F-мера, рассчитываемая для границ суперпикселей, выделенных на искажённом изображении; чем больше, тем устойчивее алгоритм).

Как правило строятся графики зависимостей данных критериев от числа суперпикселей.

4. Сравнение современных алгоритмов

Для практического сравнения различных подходов к сегментации, были выбраны три современных алгоритма из каждой группы алгоритмов:

- Алгоритм Калинина [2];
- Алгоритм ERS;
- Алгоритм SLIC.

Примеры работы данных алгоритмов приведены на рис. 2.

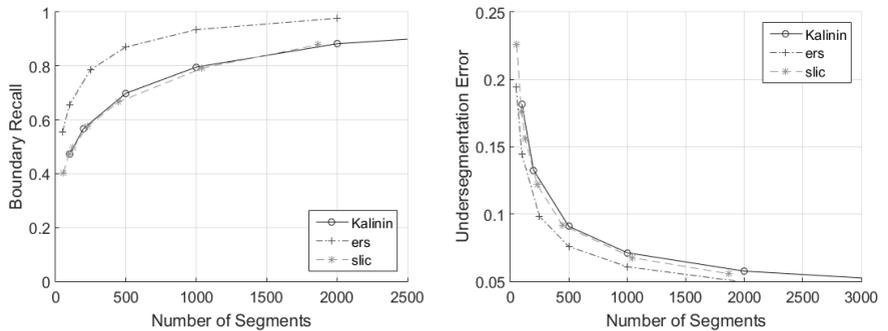


а – алгоритм Калинина, б – алгоритм ERS, в – алгоритм SLIC

Рис. 2. Примеры работы алгоритмов

Инструмент для сравнения алгоритмов реализован на языке Matlab [6] на основе изображений из тестового набора Калифорнийского университета в Беркли. Все алгоритмы реализованы на языке программирования C++.

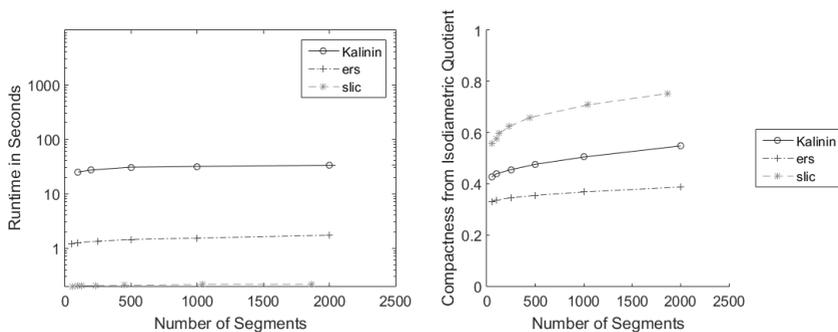
На рис. 3 приведено сравнение точности алгоритмов. Наиболее точным является алгоритм ERS, в то время как алгоритмы Калинина и SLIC показываются примерно одинаковый результат.



слева – процент обнаруженных границ,
справа – избыточность сегментации

Рис. 3. Сравнение точности алгоритмов

Из рис. 4 видно, что SLIC превосходит другие алгоритмы по времени работы и компактности получаемых суперпикселей.



слева – время работы, справа – изодиаметрический коэффициент

Рис. 4. Сравнение времени работы и компактности алгоритмов

Увеличение числа изменённых пикселей при шумах типа «соль и перец» мало сказывается на результатах алгоритмов Калинина и ERS, но сильно ухудшает результаты алгоритма SLIC (рис. 5).

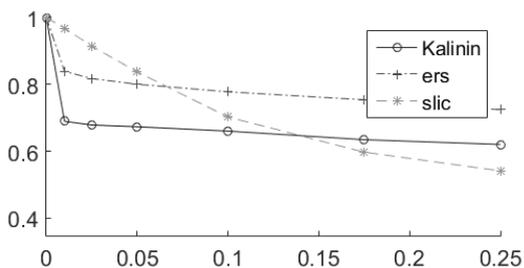


Рис. 5. Сравнение устойчивости к шуму типа «соль и перец»

Таким образом, ни один из трёх рассмотренных алгоритмов не превосходит остальные по всем критериям.

Заключение

Данная статья посвящена алгоритмам суперпиксельной сегментации. Приведены преимущества суперпиксельного представления изображений по сравнению с традиционной матрицей пикселей, которые обуславливают использование суперпиксельной сегментации на предварительных этапах обработки изображений. Рассмотрены различные подходы к суперпиксельной сегментации изображений, которая является задачей оптимизации. В ходе эксперимента, сравнивающегося несколько

современных алгоритмов суперпиксельной сегментации, продемонстрировано, что ни один из них не обладает явным преимуществом над другими, имея как сильные, так и слабые стороны.

Список литературы

1. Superpixel : Empirical Studies and Applications [Электронный ресурс] : The University of Chicago. – Режим доступа : <http://ttic.uchicago.edu/~xren/research/superpixel/>
2. Kalinin P. V. A graph based approach to hierarchical image over-segmentation / P. V. Kalinin, A. A. Sirota // Computer Vision and Image Understanding. – 2015. – Т. 130. – С. 80-86.
3. Malik J. Normalized Cuts and Image Segmentation / J. Malik, J. Shi // IEEE transactions on pattern analysis and machine intelligence. – 2000. – № 8. – V. 22. – P. 888-905.
4. Liu M. Entropy Rate Superpixel Segmentation / M. Liu [and etc.] // Conference Proceedings IEEE on Computer Vision and Pattern Recognition (Colorado Springs, 21-23 June 2011). – Colorado Springs, 2011. – P. 2097-2104.
5. Achanta R. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods / R. Achanta [and etc.] // Conference Proceedings IEEE transactions on pattern analysis and machine intelligence. – 2012. – V. 34. – № 11. – P. 2247-2281.
6. Computer Vision | Superpixel [Электронный ресурс] : Technische Universität Chemnitz. – Режим доступа : <https://www.tu-chemnitz.de/etit/proaut/forschung/cv/segmentation.html.en>

ИНТЕРАКТИВНЫЙ ИНСТРУМЕНТ ДЛЯ ВИЗУАЛЬНОГО АНАЛИЗА МНОГОМЕРНЫХ ДАННЫХ НА ОСНОВЕ ПАРАЛЛЕЛЬНЫХ КООРДИНАТ

А.А. Смолин
Студент
А.В. Сычев
Доцент

Введение

Задача разработки эффективных способов отображения разнородных наборов данных для быстрого получения человеком полезной информации до сих пор актуальна, особенно когда данные имеют многомерный характер. В последние десятилетия было предложено множество методов

визуализации данных с большим числом измерений. Однако часто они основаны на классических способах, таких, как диаграмма рассеяния, которые предназначены для отображения 2-х, максимум 3-х измерений. Поэтому при увеличении числа переменных использование таких методов визуализации затрудняется по причине их громоздкости. Несмотря на то, что они предоставляют возможность наблюдать зависимости между переменными по принципу «каждая с каждой», есть существенный риск упустить возможные корреляции на более высоком уровне [1].

Метод параллельных координат не только лишён описанных недостатков, в частности больших размеров, но и обладает некоторыми полезными свойствами. Они помогают аналитику быстро сделать вывод о характере зависимостей в многомерном наборе данных, как в целом, так и между любыми 2-я переменными. Ещё более облегчить эту задачу для человека используя алгоритмы нахождения правил, согласно которым переменные меняются в определённых диапазонах. Реализация инструмента, который соединяет в себе достоинства выбранного метода визуализации и функции предварительной фильтрации данных, позволяет максимально ускорить работу аналитика. Единственной его обязанностью в таком случае является выбор из множества найденных зависимостей тех, которые справедливы с его точки зрения, а также не были известны ранее.

1. Особенности отображения графика в параллельных координатах

Преимущества данного метода по сравнению с другими достигаются благодаря тому, что оси графика, которые соответствуют переменным визуализируемого набора данных, располагаются параллельно друг другу (и перпендикулярно [2] одной из осей) на декартовой плоскости. Этот факт обеспечивает компактность такого способа отображения многомерных данных. В целях ещё более сильного уменьшения размеров графика, расстояние между каждой из осей также может быть уменьшено, но при этом желательно, чтобы оно было одинаковым. Таким образом, анализ графика будет более достоверен [2].

Каждый экземпляр данных из визуализируемого набора отображён на графике в виде ломаной, вершины которой лежат на осях графика [3]; эти вершины и представляют собой значения переменных отдельно взятого экземпляра данных. Ломаные линии могут быть окрашены в разные цвета. Вследствие этого появляется возможность отличать экземпляры данных, принадлежащие к одному кластеру, от других (если такая информация представлена в наборе данных). Пример такого графика в параллельных координатах изображён на рис. 1 [4].

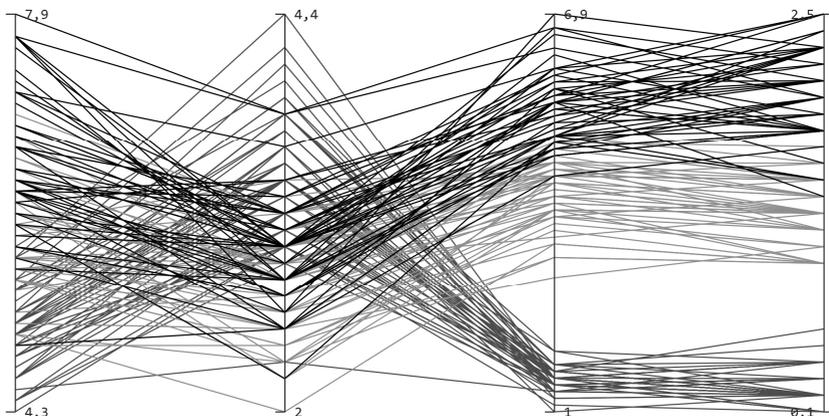


Рис. 1. График, отображающий набор данных ирисов Фишера

Учитывая описанные свойства графика, следует выделить три вида возможных проблем, которые потенциально могут помешать обнаружению важных зависимостей в наборе данных.

Во-первых, это изначально произвольный порядок расположения осей. С целью сохранения компактности графика, разумным решением здесь является возможность изменения их порядка по желанию аналитика. Во-вторых, большое число пересекающихся ломаных линий затрудняет анализ, а полезные данные могут быть скрыты некоррелирующими экземплярами. В-третьих, важную роль играет выбранный масштаб. Единичные выбросы (с сильно отличающимися от средних значениями переменных) могут значительно ухудшить отображение графика в случае использования линейных масштабов.

При разработке инструмента для работы с параллельными координатами, эффективное решение данных проблем состоит, в первую очередь, в создании интерактивного визуального компонента. Он должен быть гибким в плане сообщения с другими элементами управления, мгновенно обновляясь в зависимости от изменений, произведённых пользователем. С точки зрения функционала, основными возможностями, направленными на решение обозначенных проблем, являются изменение масштаба (непосредственно на стороне графика) и фильтрация данных (результаты этого должны отображаться на графике).

2. Алгоритм поиска ассоциативных правил как автофильтр

Дополнительной задачей являлась реализация механизма, который бы автоматически производил фильтрацию рассматриваемого набора данных по заданным аналитиком критериям. Хотя наличие ручного фильтра и

позволяет выбрать нужный ему диапазон изменения переменных, тем самым избавившись от лишних экземпляров данных, проблема поиска зависимостей между совокупностью параметров остаётся нерешённой. Для этой задачи был выбран Apriori [5] — алгоритм поиска ассоциативных правил — т. к. он:

- может работать с числовыми значениями параметров благодаря предварительной бинаризации базы данных (БД) транзакций — т. е. БД наборов элементов;

- обладает высокой скоростью поиска в наборах данных среднего размера при задании критериев в умеренных пределах.

Единичное правило представляет импликацию вида «если — то». Для каждого из них можно рассчитать вероятность появления в базе данных транзакций, называемую поддержкой [6]. Это один из тех параметров, по которому аналитик может судить о том, является ли найденное правило правилом, либо статистической погрешностью. Однако в рамках метода параллельных координат возникает вопрос, что именно следует считать элементом транзакции.

Исходя из того, что переменные в наборе данных соответствуют разнородным параметрам, за отдельный элемент принимаем каждую переменную. Однако чтобы обеспечить наличие некоторого числа диапазонов в пределах измерения, было принято решение предварительно разделять каждое из них на равное число отрезков, тем самым выделив так называемые «подтипы» элементов. Таким образом, перед процессом бинаризации, входящим в алгоритм Apriori, проводилась процедура дискретизации измерений. Имея D измерений и C кластеров, в итоге получаем $D \times C$ элементов. Тогда если в классическом случае бинаризация даст на выходе двумерную бинарную матрицу [5] размерностью $N \times E$ (где N равно числу транзакций, а E равно количеству уникальных элементов в БД транзакций), то для представления в параллельных координатах будем иметь уже трёхмерную матрицу размерностью $N \times D \times C$. Дальнейшие же процедуры полностью совпадают с оригинальным алгоритмом.

Частота дискретизация измерений, наряду с диапазоном поддержки, является ещё одним критерием, который аналитик может задавать для улучшения поиска правил в конкретном наборе данных. Более того, в паре с механизмом фильтрации, дискретизация позволяет задать точный диапазон для каждого входящего в правило измерения, тем самым исключая отображение не относящихся к правилу экземпляров данных.

3. Направления разработки приложения

Учитывая характер поставленных требований, в качестве основной технологии для разработки компонента визуализации и интерфейса была выбрана графическая подсистема Windows Presentation Foundation (WPF), входящая в состав .NET Framework. Её богатые возможности по стилизации компонентов управления с помощью языка разметки XAML позволили полностью изменить интерфейс приложения. Функции же двунаправленной привязки (binding) данных к элементам интерфейса освободили от потребности вручную следить за процессом обновления данных, как в приёмнике, так и в источнике.

Разработка проводилась по нескольким направлениям. Основным из них, как отмечалось ранее, являлось внедрение компонента, отвечающего за визуализацию графика с параллельными координатами. Так как связанный с компонентом функционал постоянно менялся, проводилась постоянная его доработка. Изначальные возможности касались масштабирования графика в нескольких режимах. В дальнейшем добавлялись функции обработки заданных пользователем параметров фильтра, как для измерений, так и для сущностей (т. е. присутствующих в наборе кластеров).

Побочное направление заключалось в реализации алгоритма Apriori для поиска ассоциативных правил, с последующим выводом найденных. Благодаря внедрённому алгоритму предварительной дискретизации и возможностям привязки данных в WPF стало возможным построить желаемую цепочку влияния компонентов интерфейса друг на друга. При выборе некоторого правила происходит выставление соответствующих параметров фильтра, что немедленно влечёт за собой обновление внешнего вида графика.

На протяжении всего цикла разработки также производилось: изменение внешнего вида стандартных элементов интерфейса; создание дополнительных элементов, отсутствующих в стандартном наборе WPF; сохранение новых параметров приложения в файл конфигурации; постоянное дополнение русской и английской локализации и т. д.

4. Возможности итогового инструмента

Полученный инструмент (рис. 2) позволяет работать со структурированными текстовыми файлами форматов DAT и CSV, которые используют разделители для параметров (так называемый Delimiter-Separated Values). Набор основных его функций можно разделить на 3 категории:

- внешний вид и масштабирование графика;
- фильтрация данных;
- поиск ассоциативных правил.

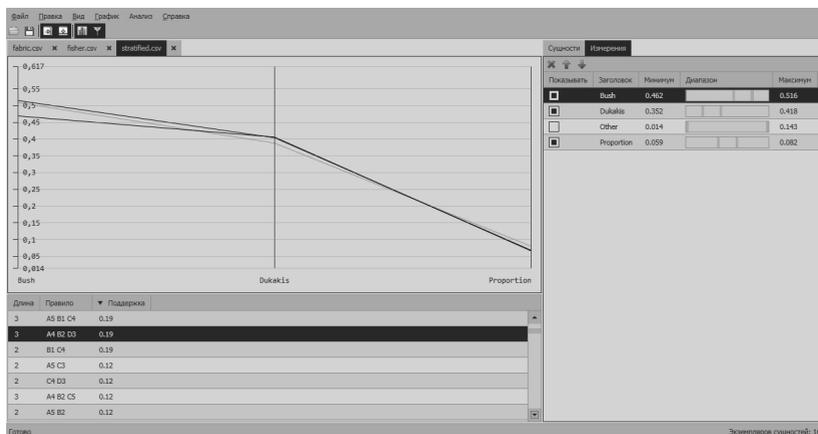


Рис. 2. Интерфейс главного окна приложения

В первую категорию входят функции включения и отключения на графике шкал, подписей к измерениям, а также легенды; к возможностям масштабирования относится опция «Единые пределы осей», в случае отключения которой выставляются предельные диапазоны для каждой из осей, а также опция переключения между линейным (выбранным по умолчанию) и логарифмическим масштабом.

Ко второй категории относится функционал, который активируется путём выбора опции «Фильтр» из меню «График». Если она отключена, то изменение диапазонов измерений на соответствующей вкладке боковой панели (на рис. 2 справа) лишь изменяет масштаб осей. В противном же случае происходит непосредственная фильтрация экземпляров данных, не удовлетворяющих заданным критериям. Если хотя бы одна переменная из экземпляра не входит в установленный для неё диапазон, то экземпляр полностью отбрасывается, т. е. его отображения на графике не происходит. Такое поведение позволяет потенциально исключить все экземпляры, в которых аналитик не заинтересован. Также с помощью фильтра можно целиком отключать на графике измерения и сущности (кластеры), и что ещё более важно — имеется возможность изменения порядка следования осей.

Третья категория тесно связана со второй. Поиск ассоциативных правил в открытом наборе инициируется выбором одноимённого пункта меню «Анализ». От пользователя требуется предварительно указать два параметра в диалоговом окне:

– диапазон поддержки, варьирующийся от 0,01 до 1. Чем шире этот диапазон, тем больше правил будет найдено, однако время поиска также увеличивается;

– частота дискретизации (в каждом измерении); от 2-х до 10-и. Большие значения параметра, в совокупности со многим числом измерений, также отрицательно влияют на быстроту поиска.

После этого все найденные правила отображаются в нижней панели (на рис. 2 под графиком). При выборе какого-либо из них происходит мгновенная фильтрация, после чего возможно наблюдать данное правило как в графической интерпретации — в параллельных координатах — так и в виде диапазонов, подходящих под него.

Заключение

Имеющийся в инструменте функционал является необходимым для быстрого получения достоверных результатов при визуальном анализе графика с параллельными координатами. При этом, чем больше экземпляров данных в исходном наборе, тем сильнее потребность в использовании функций фильтрации и масштабирования.

Разработанный инструмент решает перечисленные ранее проблемы, которые становятся заметны при детальном рассмотрении метода параллельных координат. Во-первых, ухудшение внешнего вида графика вследствие наличия выбросов в значениях переменных действительно можно решить использованием логарифмического масштаба, поскольку логарифм отношения больших величин растёт намного медленнее, чем их разность. Во-вторых, ручная фильтрация позволяет избавиться от лишних данных, не затрагивая интересные; достаточно лишь соответствующим образом задать диапазоны переменных, которые не коррелируют у искомым и фильтруемым экземпляров данных. Наконец, в-третьих, поиск ассоциативных правил освобождает аналитика от кропотливого поиска неочевидных зависимостей. Единственным вопросом, на который он должен будет ответить, является адекватность рассматриваемого им правила.

Все используемые методы фильтрации теоретически применимы для любых объёмов данных. Возможная проблема состоит лишь в возросшем времени на их обработку. В этом случае увеличение скорости может быть достигнуто только путём тщательной оптимизации каждого алгоритма, применяющегося при выполнении трудоёмкой операции.

Список литературы

1. Wegman E.J. Hyperdimensional Data Analysis Using Parallel Coordinates / E.J. Wegman // Journal of the American Statistical Association. – 1990. – Vol. 85 (411). – P. 664-675.

2. Parallel Filter : A Visual Classifier Based on Parallel Coordinates and Multivariate Data Analysis / Y. Xu, W. Hong, N. Chen and others // Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence : Third International Conference on Intelligent Computing (Qingdao, China, 21-24 August 2007). – Berlin : Springer Berlin Heidelberg, 2007. – Vol. 4682. – P. 1172-1183.
3. Inselberg A. Parallel Coordinates: Visual Multidimensional Geometry and its Applications [Electronic resource] / A. Inselberg. – 1992-2004. – Access mode : http://www.math.tau.ac.il/~aiisreal/index_files/lect-pdf/lect-2d.pdf
4. Koch I. Analysis of Multivariate and High-Dimensional Data / I. Koch. – New York : Cambridge University Press, 2014. – 504 p.
5. Шахиди А. Apriori — масштабируемый алгоритм поиска ассоциативных правил [Электронный ресурс] / А. Шахиди // BaseGroup Labs : Технологии анализа данных. – 2004. – Режим доступа : <https://basegroup.ru/community/articles/apriori>
6. Tan P.-N. Introduction to Data Mining / P.-N. Tan, M. Steinbach, V. Kumar. – London : Pearson, 2005. – 769 p.

ОЦЕНКА УРОВНЯ ИНФОРМАЦИОННОЙ ЗАЩИЩЕННОСТИ ОРГАНИЗАЦИИ НА ОСНОВЕ БАЙЕСОВСКИХ СЕТЕЙ ДОВЕРИЯ

С.В. Смотров
Студент
В.В. Гаршина
Доцент

Введение

Одной из наиболее ценных задач для всех организации считается обеспечение её информационной безопасности. Такие параметры, как конфиденциальность, целостность, доступность информационных ресурсов обуславливают не только качество и оперативность принятия технических решений, но и эффективность их реализации.

Деятельность медицинского учреждения рискованна без решения задач безопасности. Многие процессы находятся в прямой зависимости от того, как работает информационная инфраструктура, то есть вычислительные комплексы, программные и аппаратные средства, а также персонал, их использующий. Если все это начинает давать сбои, например, за счет заражения компьютеров вирусами или атак хакеров, последствия для медицинского учреждения могут быть очень плачевными. Больница

© Смотров С.В., Гаршина В.В., 2016

может утратить базы данных клиентов, работников, диагнозов, лекарств и другую необходимую информацию, собираемую годами, лишиться средств и потерять доверие пациентов.

Целью данной работы является анализ и разработка системы для оценки уровня информационной защищенности медицинского учреждения на базе байесовских сетей доверия.

1. Факторы безопасности для информационной системы медицинской организации

Объектом исследования в данной работе является защита информации в одном из медицинских учреждений Воронежской области. Предметом исследования являются процессы и средства защиты информации.

Информационной базой исследования стали законодательные акты и нормативно – правовая база в области защиты информации, научные и методические разработки экспертов.

В ходе исследования были проанализированы проблемы и систематизированы существующие методы и средства обеспечения информационной безопасности в медицинских организациях. Выделены объекты информационной безопасности, мероприятия по защите данных объектов и нейтрализации возможных угроз. А также, проанализированы существующие методики оценки параметров уровня информационной безопасности медицинских учреждений.

Классификация угроз информационной безопасности медицинского учреждения [1]:

- по природе возникновения (естественные и искусственные);
- по степени преднамеренности проявления (угрозы случайного характера – вызваны ошибками или халатностью персонала, некомпетентностью; неумышленная порча оборудования, удаление, искажение файлов; угрозы преднамеренного действия);
- по непосредственному источнику угроз (угрозы, источником которых является природная среда; угрозы, непосредственным источником которых является человек);
- по положению источника угроз (угрозы, источник которых расположен вне территории; угрозы, источник которых расположен в пределах территории медицинского учреждения);
- по степени воздействия на автоматизированную систему (пассивные угрозы – при реализации они ничего не меняют в структуре и содержании; активные угрозы – при воздействии вносят изменения в структуру и содержание автоматизированной системы);
- по способу доступа к ресурсам АС (до авторизации, либо после).

2. Байесовские сети доверия как инструмент моделирования системы безопасности информации

Байесовские сети доверия (БСД) как инструмент моделирования применяются в предметных областях, характеризующихся неопределенностью и случайностью [2]. Эти особенности моделируемых систем могут появляться по причинам:

- сложность предметной области;
- недостаточность описания предметной области, в силу отсутствия информации;
- влияние случайных факторов.

Для байесовских сетей доверия также применяется другое определение – причинно-следственная сеть.

Аппарат БСД при построении модели использует субъективные вероятности, представляющие собой экспертные оценки априорных и апостериорных условных вероятностей событий, включенных в модель. Субъективную вероятность можно рассматривать как меру уверенности эксперта в возможности наступления события.

Рассмотрим пример сети (рис. 1), в которой вероятность пребывания вершины «e» в различных состояниях (e_k) зависит от состояний (c_i, d_j) вершин «c» и «d» и определяется выражением (формула распределения вероятностей):

$$p(e_k) = \sum \sum p(e_k | c_i, d_j) * p(c_i d_j), \quad (1)$$

где $p(e_k | c_i, d_j)$ – вероятность пребывания в состоянии e_k в зависимости от состояний c_i, d_j .

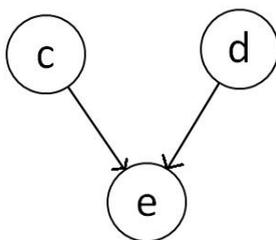


Рис. 1. Пример простейшей байесовской сети доверия

3. Инструмент моделирования

Для разработки системы поддержки принятия решений (СППР) информационной защищенности организации в данной работе использовался Hugin (www.hugin.com; Hugin AIS, Дания). Hugin является

программой реализацией системы принятия решений на основе Байесовских сетей доверия [3].

Система Hugin позволяет создавать системы поддержки принятия управленческих решений в условиях неопределенности на основе моделей проблемной области. Система ориентирована на построение моделей на основе теории сетей Байеса и диаграмм влияния.

4. Структура байесовской сети доверия для СППР в области защиты информации деятельности больницы

После проведения анализа деятельности медицинского учреждения была разработана модель БСД для обеспечения защиты информации больницы (рис. 2).

В построении модели экспертной системы были выделены основные вершины:

- А – методы программного характера;
- В – методы технического характера;
- С – методы организационного характера;
- X – угроза безопасности информации больницы;

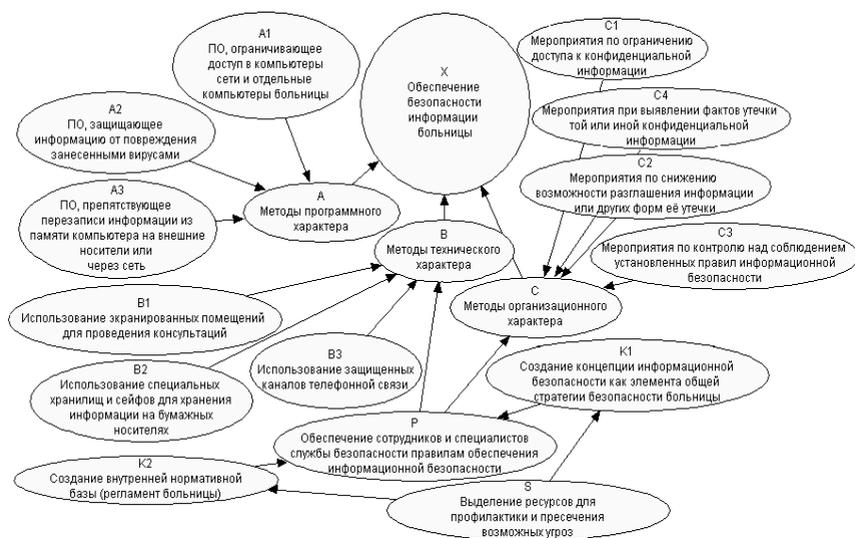


Рис. 2. БСД информационной защищенности организации

5. Обучение БСД

Дальнейшее улучшение качества прогнозирования защищенности больницы может быть достигнуто путем обучения Байесовской сети на

6. Разработка СППР

С помощью Hugin .Net API создано приложение, позволяющее открывать вероятностные графические модели, созданные в Hugin и производить перерасчет вероятностей, без изменения самой модели. Функции приложения:

- Редактирование любой вершины;
- Изменение вероятности для любой вершины;
- Произведение пересчета вероятностей вершин при изменении вероятности для одной (производится автоматически).

Приложение позволяет, изменяя состояние отдельной вершины, проанализировать – как меняются другие вершины.

Используя входные данные – субъективные вероятности экспертов предметной области, мы имеем для вершины А (методы защищенности программного характера): yes – 58,5; no – 41,5.

Теперь понаблюдаем, как изменяется общая информационная безопасность при $A1 = 100.00$, где А1 это вершина, обозначающая один из методов программного характера, а именно – ПО, ограничивающее доступ в компьютеры сети и отдельные компьютеры медицинского учреждения. Вершина А стала принимать значения: yes – 63,03; no – 36,97. Пересчет вероятностей произведен, результат мы можем увидеть на скриншоте приложения (рис. 4).

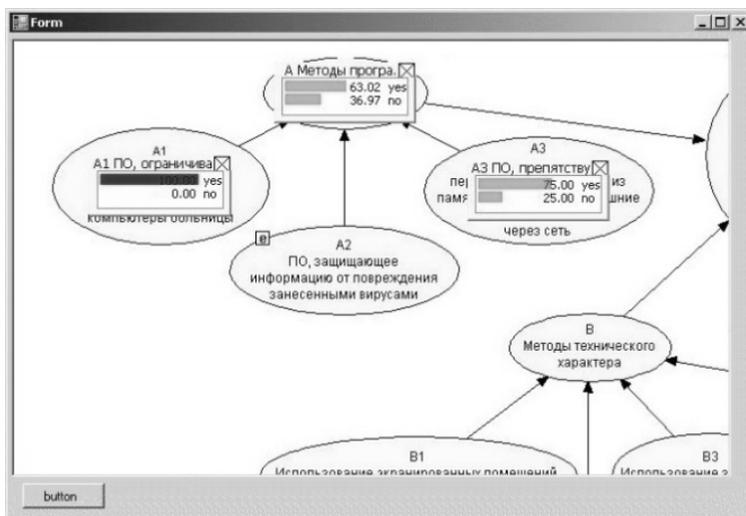


Рис. 4. Интерфейс приложения в режиме изменения вероятностей

Заключение

В ходе выполнения работы были решены следующие основные задачи:

- Проведен анализ предметной области, выделена модель информационных потоков медицинского учреждения;
- Проанализированы типы возможных информационных угроз, проведена оценка возможных каналов утечки конфиденциальной информации;
- Определены основные методы защиты информации;
- Построена вероятностная модель для оценки надежности системы защиты информации на основе БСД;
- Реализована СППР на основе построенной модели.

Разработанная СППР даёт возможность быстро оценить угрозу безопасности информации в медицинском учреждении и принять решения для дальнейших результативных действий по защите.

Список литературы

1. Безбогов А.А. Безопасность операционных систем: учебное пособие / А.А. Безбогов, А.В. Яковлев, Ю.Ф. Мартемьянов. – М. : Издательство Машиностроение-1, 2007. – 220 с.
2. Тулупьев А.Л. Байесовские сети: Логико-вероятный подход / А.Л. Тулупьев [и др.]. – СПб. : Наука, 2006. – 607 с.
3. Поллак Р.А. Инструментальные средства разработки экспертных систем : Учебное пособие. – Челябинск : Изд. ЮУрГУ, 2003. – 65 с.
4. Hugin GUI Help [Электронный ресурс] : база данных. – Режим доступа : <http://download.hugin.com/webdocs/manuals/Htmlhelp>

ПОСТРОЕНИЕ СВЕРХРАЗРЕШЕНИЯ ПО ОДНОМУ ИЗОБРАЖЕНИЮ

Д.М. Суходолов

Студент

А.А. Крыловецкий

Доцент

Введение

В современном мире всё чаще возникает необходимость получить наиболее точные и качественные данные [1, 2]. Причём, даже если

используется самая лучшая на данный момент техника, все равно иногда возникают ситуации, когда необходимо увеличить отдельную часть изображения [3, 4, 5].

Одним из вариантов повышения качества исходных фотографий является использование алгоритмов получения сверхразрешения изображений. Задачей данного класса методов является восстановление изображений более высокого разрешения из одного или более изображений низкого разрешения, полученных на некоторой сцене [6, 7].

В большинстве работ, изучающих сверхразрешение изображений, рассматривается получение сверхразрешения из нескольких исходных изображений одной сцены [1, 8]. Однако существуют случаи, когда можно получить только одно исходное изображение. Например, в процессе рентгенографии на тело человека воздействует вредное рентгеновское излучение, и делать часто повторные снимки является крайне нежелательным [3]. В случае же оцифровки раритетной книги или картины их структура будет постоянна, даже если сделать несколько высококачественных снимков, поэтому каждое новое изображение в серии изображений не производит никакой новой информации об исходной сцене [4, 5].

Изображения часто имеют похожие части. Например, сектор изображения размером 5×5 пикселей часто повторяется в изображении несколько раз, как в исходном размере изображения, так и в других его размерах (рис. 1). Исследования повторяемости секторов показали, что существует значительная избыточность секторов в пределах одного изображения и различных его размерах. Так, в среднем, более чем 90% секторов в изображении имеет 9 или более похожих секторов [9]. Степень сходства секторов оценивается с помощью метода К ближайших соседей.



*а – при исходном размере изображения,
б – при различных размерах изображения*

Рис. 1. Повторяемость секторов в изображениях различных размеров

1. Методы получения сверхразрешения изображений

Существует два метода получения сверхразрешения: классическое сверхразрешение и основанное на самообучающейся системе с базой образцов [6].

Классическое сверхразрешение предполагает, что имеется набор изображений низкого разрешения L_1, \dots, L_n некоторой исходной сцены, пиксели которого соответственно согласовываются друг с другом с незначительными отклонениями [1, 10].

Принято считать, что изображение с низким разрешением L является результатом свёртки изображения высокого разрешения H с ядром свёртки B_j , прошедшим затем операцию децимации, которая привела к уменьшению размера изображения:

$$L_j = (H * B_j) \downarrow s_j, \quad (1)$$

где \downarrow обозначает операцию децимации, s_j - коэффициент уменьшения масштаба между H и L_j .

Поскольку свёртка является линейной операцией, уравнение (1) порождает набор линейных ограничений на изображении H . Для большого количества исходных изображений получается несколько уравнений для каждого пикселя и задача становится решаемой.

Метод получения сверхразрешения с базой образцов подразумевает то, что соответствия между секторами изображений низкого и высокого разрешения становятся известны из заранее созданной базы образцов, содержащей в себе образцы пар секторов низкого и высокого разрешения изображений. Найдя в базе подобный необходимому сектор среди образцов секторов низкого разрешения, можно получить соответствующий ему сектор в изображении высокого разрешения [11].

2. Объединение методов получения сверхразрешения для одного исходного изображения

Рассмотрим возможность объединения данных методов для получения сверхразрешения в условиях наличия только одного исходного изображения [9]. Для увеличения разрешения изображения в два раза создаётся пирамида изображений из 7 уровней (рис. 2).



Рис. 2. Пирамида изображений

Положим $L = I_0$ – входное изображение низкого разрешения, находящееся на уровне 4. Пусть далее I_{-s} обозначает такую версию L после децимации, что I_0 – это увеличение I_{-s} в sX раз. Аналогично положим I_s как увеличение I_0 в sX раз. Для каждого сектора P_j алгоритм находит k его ближайших соседей во всех изображениях $I_0, I_{-1}, \dots, I_{-n}$. Совпадения в пределах I_0 называются совпадениями в исходном разрешении, а совпадения в I_{-1}, \dots, I_{-n} – совпадениями в других разрешениях исходного изображения.

Каждый ближайший сектор P_j , найденный в исходном разрешении входного изображения, можно рассматривать как наблюдение P_j на сцене с небольшими сдвигами. В данном случае можно говорить об основании использования метода получения классического сверхразрешения. Точно так же, ближайшие сектора P_j , найденные в других разрешениях исходного изображения, делают применимым подход для сверхразрешения с базой образцов. Если находится достаточное количество секторов, то уравнение (1) будет полностью определено для каждого отдельного пикселя.

Что касается ядра размытия B_l на уровне I_l , то при равномерном масштабном коэффициенте $s_l = \alpha^l$ гарантируется, что если два изображения в пирамиде найдены на расстоянии m уровней друг от друга (например, I_l и I_{l+m}), то они будут связаны одним и тем же ядром размытия B_m вне зависимости от l . Оптимально выбрать $\alpha = 2^{\frac{1}{3}}$ [9].

В данном подходе множество линейных уравнений было решено не сразу для уровня H , а решалось постепенно, от наименьшего по разрешению неизвестного уровня пирамиды к наибольшему. Так, когда решаются уравнения для изображения I_{l+1} , используются не только соответствия между секторами изображений низкого и высокого разрешения, полученные из входного изображения L , но и все, ставшие известными на прошлых шагах соответствия секторов из новых восстановленных изображений I_0, \dots, I_l . Данный процесс повторяется, пока не будет достигнут уровень разрешения H . При решении используются только чёрно-белые изображения, что позволяет с большей вероятностью обнаружить похожие секторы.

Выполняя действия по получению сверхразрешения, одним из необходимых условий успешной работы зачастую является сохранение исходных цветов фотографии. В целях упрощения данной задачи производится переход из цветовой модели RGB в YIQ. Таким образом, можно выполнить все действия алгоритма с яркостной составляющей Y, а искусственные цветоразностные составляющие I и Q увеличить до необходимого разрешения методами кубической интерполяции. После этого выполнить обратный переход в цветовую модель RGB.

3. Полученные результаты

Для оценивания степени сходства секторов в разработанном программном модуле используется метод К ближайших соседей, который возможно модифицировать с помощью различных метрик и методов поиска. В ходе исследования были изучены следующие метрики: Евклидова метрика, Манхэттенское расстояние, Расстояние Махаланобиса и Расстояние Хэмминга [12]. Среди методов поиска рассматривался метод полного перебора согласно алгоритму и метод К-мерного дерева.

В случае сравнения сверхразрешения и методов кубической интерполяции, время, необходимое на работу кубической интерполяции, затрачивается значительно меньшее, чем на методы сверхразрешения. Однако для восстановления исходного изображения, как правило, первостепенным показателем является качество полученного изображения, а потом уже время выполнения. Сравнение представлено на рис 3.



а



б

а – кубическая интерполяция, б – сверхразрешение изображений

Рис. 3. Сравнение кубической интерполяции и сверхразрешения изображений

При рассмотрении множества примеров было выявлено, что из методов поиска наиболее целесообразно использовать метод К-мерного дерева, так как он позволяет получить хорошие результаты за значительно более приемлемое время при использовании большинства исследованных метрик.

Среди метрик с наилучшей стороны себя показали Евклидово расстояние и Манхэттенская метрика. Используя их, получается наилучшее качество за наилучшее время, по сравнению с другими метриками. Расстояния Махаланобиса и Хэмминга показали себя как совершенно неприменимые к задаче с позиции затраченного времени и, что самое существенное, с точки зрения получаемых результатов. Таким образом, их применение в данной области не рекомендуется.

Заключение

Результаты сравнения метода повышения разрешения изображений с помощью кубической интерполяции и сверхразрешения изображений показывают очевидное преимущество и работоспособность метода получения сверхразрешения изображений из одного изображения.

Список литературы

1. Baker S. Limits on Super-Resolution and How to Break Them / S. Baker, T. Kanade // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2000. – № 24. – P. 1167-1183.

2. Buades A. A non-local algorithm for image denoising / A. Buades // Computer Vision and Pattern Recognition. – 2005. – V. 2. – P. 60-65.
3. Oppelt A. Imaging Systems for Medical Diagnostics : Fundamentals, Technical Solutions and Applications for Systems Applying Ionizing Radiation, Nuclear Magnetic Resonance and Ultrasound / A. Oppelt. – Erlangen : Publicis, 2006. – 996 p.
4. Bulow A. Preparing Collections for Digitization / A. Bulow, J. Ahmon. – London : Facet Publishing, 2010. – 192 p.
5. Perrin J. Digitizing Flat Media : Principles and Practices / J. Perrin. – Lanham : Rowman & Littlefield Publishers, 2015. – 156 p.
6. Milanfar P. Super-resolution imaging / P. Milanfar. – Boca Raton : CRC Press, 2011. – 492 p.
7. Chaudhuri S. Super-resolution imaging / S. Chaudhuri. – Mumbai : Kluwer Academic Publishers, 2002. – 294 p.
8. Сирота А.А. Блочные алгоритмы обработки изображений на основе фильтра Калмана в задаче построения сверхразрешения / А.А. Сирота, А.Ю. Иванков // Компьютерная оптика. – 2014. – Т. 38. – № 1. – С. 118-127.
9. Glasner D. Super-Resolution from a Single Image / D. Glasner, S. Bagon, M. Irani // Computer Vision : 2009 IEEE 12th International Conference on (Kyoto, 27 Sept. 2009-4 Oct. 2009). – Kyoto, 2009. – P. 349-356.
10. Irani M. Improving resolution by image registration / M. Irani, S. Peleg // Graphical Models and Image Processing. – 1991. – Vol. 53. – № 3. – P. 231-239.
11. Freeman W. Learning low-level vision / W. Freeman, E. Pasztor, O. Carmichael // International Journal of Computer Vision. – 2000. – V. 1. – P. 47-35.
12. Колмогоров А. Н. Элементы теории функций и функционального анализа / А. Н. Колмогоров, С. В. Фомин. – М. : Физматлит, 2006. – 543 с.

ПРОЕКТИРОВАНИЕ СЕТЕЙ ПОДВИЖНОЙ СВЯЗИ С КODOVЫМ РАЗДЕЛЕНИЕМ КАНАЛОВ

С.И. Тюленев

Студент

Н.Н. Винокурова

Доцент

Введение

Сеть подвижной связи, или как её ещё называют сеть сотовой связи, представляет собой вид мобильной радиосвязи, в основе которой лежит сотовая сеть. Главная особенность состоит в том, что общая зона покрытия делится на ячейки, которые имеют форму сот, перекрывают друг друга и образуют сеть. На ровной поверхности зона покрытия одной базовой станции представляет собой окружность, поэтому составленная таким образом сеть, имеет вид шестиугольных ячеек, т. е. сот.

Сеть состоит из разнесённых в пространстве приёмопередатчиков, которые работают в одном частотном диапазоне, а также коммутирующее оборудование, позволяющее определить текущее местоположение подвижных абонентов и обеспечивать непрерывность связи из зоны действия одного приёмопередатчика в зону действия другого при перемещении абонента.

Для того чтобы сигналы не накладывались и не создавали помех друг другу, существуют технологии разделения каналов: частотное (FDMA), временное (TDMA), пространственное (SDMA), поляризационное (PDMA) и кодовое (CDMA) разделение каналов [1].

1. CDMA

В последние годы значительный прогресс, в телекоммуникационных технологиях достигнут благодаря переходу на цифровые виды связи, которые, в свою очередь, базируются на стремительном развитии микропроцессоров. Один из ярких примеров этого – появление и быстрое внедрение технологии связи с цифровыми шумоподобными сигналами и с методами многостанционного доступа с кодовым разделением каналов. Технология сотовой (подвижной) связи с кодовым разделением каналов CDMA (Code Division Multiple Access) в последнее время составляет серьёзную конкуренцию цифровым технологиям, таким как GSM.

В системах с кодовым разделением каналов каждый голосовой поток отмечен своим уникальным кодом и передается на одном канале одновременно со многими другими кодированными голосовыми потоками

на одной частоте и в один и тот же промежуток времени. Принимающая сторона использует тот же код для выделения сигнала из шума. Единственное отличие между множественными голосовыми потоками это уникальный код. Канал, как правило, очень широк и каждый голосовой поток занимает целиком всю ширину диапазона. В системах CDMA могут устанавливаться очень прочные и защищенные соединения, несмотря на экстремально низкую величину мощности сигнала, теоретически – сигнал может быть слабее, чем уровень шума [2].

2. Цель эксперимента

Цель эксперимента в данной статье – исследование влияния аддитивного белого гауссовского шума на передачу сигналов с кодовым разделением каналов.

Для этого в среде Matlab была смоделирована система, полностью повторяющая работу CDMA. Передается последовательность бит, на которую накладывается аддитивный белый гауссовский шум. Так как сигнал является шумоподобным, то для искажения необходимая величина помехи должна быть в несколько раз больше сигнала.

Схема генерации сигнала представлена на рис 1.

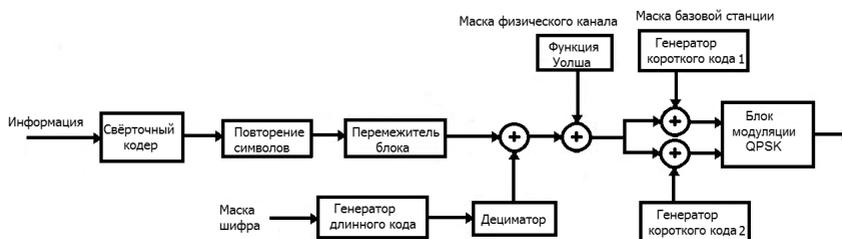


Рис. 1. Схема генерация сигнала на базовой станции

Сигнал (информационная последовательность) поступает на блок помехоустойчивого кодирования. Далее происходит повторение символов для борьбы с одиночными ошибками. После повторителя сигнал поступает в блок перемежения сигнала. Блок предназначен для борьбы с пачками ошибок в эфире. Пачки ошибок – искажение нескольких бит информации подряд. Принцип такой. Поток данных записывается в матрицу по строкам. Как только матрица заполнена, начинаем с нее передавать информацию по столбцам. Далее сигнал поступает в блок кодирования (от подслушивания). На информацию накладывается маска длиной 42 бита. Эта маска является секретной. При несанкционированном перехвате данных в эфире невозможно декодировать сигнал, не зная маски. Далее сигнал умножается на функцию Уолша для физического разделения

каналов, т.е. язык общения из примера с рестораном. Далее следует блок перемножения сигнала на две М-функции или еще их называют ПСП – псевдослучайными последовательностями. Блок предназначен для перемешивания сигнала для блока модуляции. Каждой назначенной частоте назначаются разные М-функции. После этого используется квадратурная фазовая модуляция (рис. 1).

3. Разделение каналов

Для того чтобы создать множество сигналов необходимо использовать фазовые различия. Поэтому при построении такого рода сигналов используют радиоимпульсы с двумя значениями фазы 0 и π радиан. Условно импульс с фазой 0 обозначается "1", с фазой π – "-1". Для этого используется функция Радемахера [3].

Функция Радемахера — кусочно-постоянная периодическая функция, принимающая только два значения 1 и -1 на всей области определения. Введены Гансом Радемахером в 1922 году. График функции представляет собой меандр. Функция Радемахера может быть выражена следующим образом:

$$\text{rad}_n(t) = \text{sign}[\sin(2^n)\pi t], \quad (1)$$

где $0 \leq t \leq 1$, $n = 0, 1, 2 \dots$

Все функции Радемахера являются нечетными относительно середины интервала и, следовательно, не могут быть использованы для аппроксимации сигналов $s(t)$, четных относительно момента $t=1/2$. Иными словами, система функций Радемахера является неполной.

Функции Уолша образуют полную ортонормированную систему и могут быть получены путём перемножения степеней соответствующих функций Радемахера.

Функция Уолша может быть упорядочена по Пэли, Уолшу и Адамару. Наилучшим является упорядочение по Пэли, наихудшим по Адамару. Упорядочение зависит от расположения спектральных компонентов [3]. По Пэли они расположены так, что каждый следующий по модулю, не больше предыдущего (рис. 2). Функция Радемахера может быть выражена следующим образом:

$$\text{Pal}_n(t) = \text{rad}_{m+1}\text{rad}_{n+1}\text{rad}_{i+1}, \quad (2)$$

Используя функции Уолша можно составить матрицу Уолша–Адамара упорядоченную по Пэли.

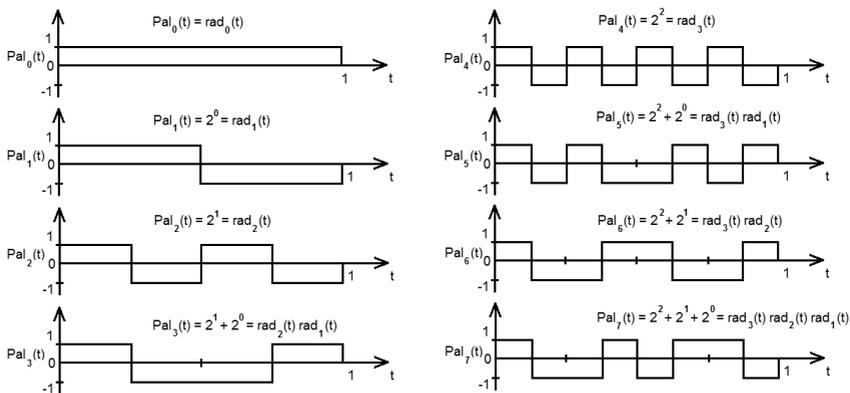


Рис. 2. Первые восемь функций Уолша упорядоченные по Пэли

4. Влияние помехи при передаче

Для начала следует найти такую величину помехи, при которой будет происходить искажение одного бита информации. Для передачи была взята следующая последовательность бит: 10101010.

Для первого эксперимента была взята узкополосная помеха, заданная по нормальному распределению (Гаусс-распределение). После модуляции сигнала помеха суммировалась с передаваемым сигналом (аддитивная помеха). Отношение сигнал/шум равно 2 (рис. 3). Данные представлены в виде кусочно-постоянной функции.

Искажение одного бита начало происходить при достижении значения сигнал/шум равное величине 1/12. После проведения 100 испытаний было замечено, что вероятность искажения одного бита, составляет 56%. В 44 из 100 испытаниях искажения последовательности бит не происходило.

При достижении значения сигнал/шум равное 1/15 начало происходить искажения до 2-х бит. Из 100 проведенных испытаний в 18 искажений бит не происходило, в 44 произошло искажение одного бита, в 38 искажение двух бит.

Значение сигнал/шум было увеличено до значения 1/16. Из 100 проведённых испытаний в 9 искажение бит не происходило, в 25 произошло искажение одного бита, в 54 испытаниях произошло искажение двух бит, в 12 испытаниях исказились 3 бита. При уменьшении соотношения сигнал/шум вероятность искажения трёх бит стало быстро увеличиваться. При значении 1/18 происходит искажение до 4 бит. Дальнейшие испытания стали нецелесообразные, так как происходит искажения половины передаваемых бит и в этом случае реальные системы передают сигнал повторно.

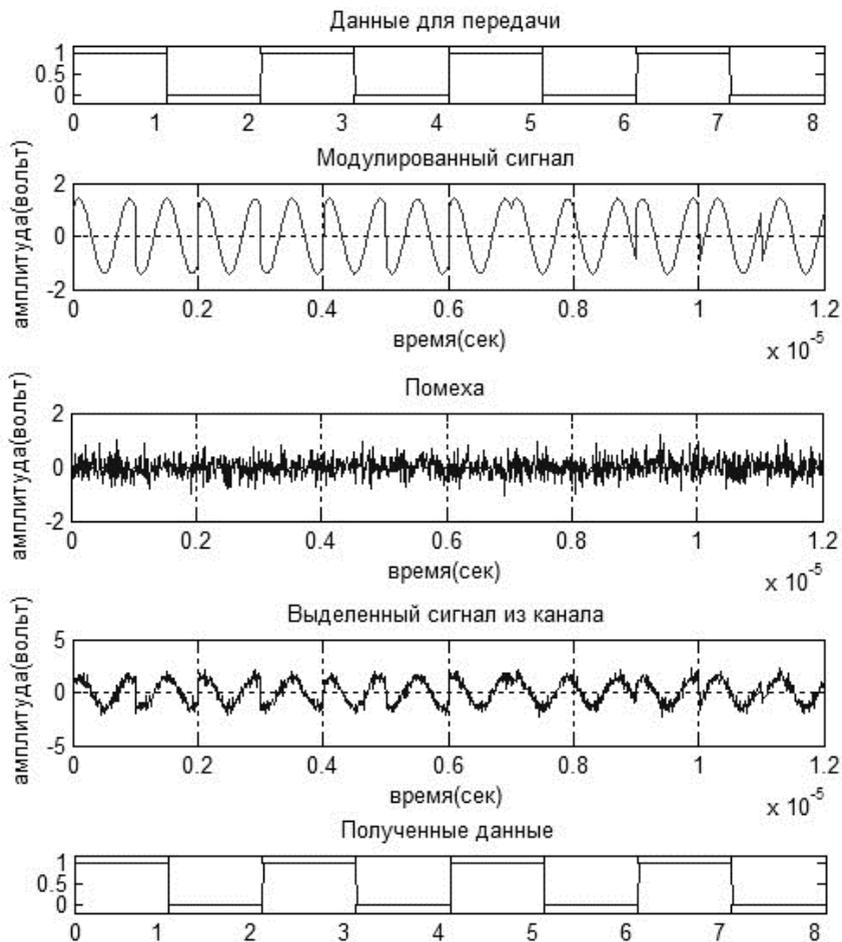


Рис. 3. Отношение сигнал/шум равное 2

Заключение

Данная статья посвящена системам с кодовым разделением каналов (CDMA) и влиянию аддитивного белого гауссовского шума на такие системы. В итоге данного эксперимента были получены значения сигнал/шум, при которых происходит искажения до одного, двух, трёх и четырёх передаваемых бит. Так как сигнал в данной системе является шумоподобным, то величина помехи в несколько раз превысила величину сигнала. Таким образом, можно сделать вывод, что системы подвижной

связи являются помехоустойчивыми к аддитивным помехам. В настоящее время в CDMA применяются мощные и совершенные алгоритмы обнаружения и коррекции ошибок. В случае искажения бит данных они или исправляются, или обрабатываются таким образом, чтобы искажения были минимальными.

Список литературы

1. Системы связи с кодовым разделением каналов (CDMA) : учебник / В.Ю. Бабков [и др.] ; отв. ред. В.Ю. Бабков. – 3-е изд., перераб. и доп. – СПб : Издательский центр Санкт-Петербургского государственного университета телекоммуникаций им. Бонч-Бруевича, 2004. – 120 с.
2. Сакалема Д.Ж. Сеть подвижной связи CDMA. Построение и проектирование / Д.Ж. Сакалема. – М. : Science Press, 2012. – 240 с.
3. Никитин Г.И. Применение функций Уолша в сотовых системах связи с кодовым разделением каналов / Г.И. Никитин. – СПб : Издательский центр Санкт-Петербургского государственного университета аэрокосмического приборостроения, 2003. – 86 с.

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ HADOOP ДЛЯ АНАЛИЗА "БОЛЬШИХ ДАННЫХ" НА ПРИМЕРЕ ЗАДАЧИ ГЕОАНАЛИТИКИ

А.Г. Чурсин

Студент

Н.А. Тюкачев

Доцент

Д.И. Соломатин

Старший преподаватель

Введение

Данная работа посвящена анализу больших массивов данных с использованием парадигмы MapReduce и стека технологий Hadoop на примере задачи для крупной телекоммуникационной компании, поставщика услуг сотовой связи и широкополосного интернета, в рамках проекта Hadoop for Big Data.

Оператор связи может знать местоположение абонентов. Например, известно, что в такой-то зоне часто появляются люди с iPhone 6, возможно они пользуются стандартом связи 4G, а значит, нужно построить в этой соте LTE-базовую станцию; в другом месте хватит 2G.

Строго говоря, Big Data – это обработка огромных массивов данных, но, например, работа с базой данных под управлением СУБД Oracle на 20 Гб или 4 Гб – это ещё не Big Data, это просто высоконагруженная база данных или highload-БД. Ключевое отличие Big Data от обычных высоконагруженных систем состоит в возможности строить гибкие запросы при обработке данных любого объема.

Одним из самых известных и обсуждаемых проектов в области обработки больших данных является Hadoop, разрабатываемый фондом Apache Software Foundation. Он представляет собой свободно распространяемый набор из утилит и библиотек для разработки и выполнения программ распределенных вычислений [1].

MapReduce – это модель распределенной обработки данных, предложенная компанией Google для обработки больших объёмов данных на компьютерных кластерах [1].

1. Формулировка постановки задачи

Необходимо сформировать единовременную выгрузку, определив количество уникальных абонентов, которые находились в течение рабочего дня в каждой ячейке зоны разбиения города Нижний Новгород за период с 14 по 20 сентября 2015 года для оптимального расположения офисов монобренда.

Отчет формируется относительно всех зон разбиения (zone id) в дальнейшем именуемых для краткости zid.

Каждые сутки делятся на временные интервалы:

- 00.00.00-05.59.59;
- 06.00.00-08.59.59;
- 09.00.00-20.59.59;
- 21.00.00-23.59.59.

В поле total (суммарное количество) записывается число абонентов, удовлетворяющих следующему условию. Существует принадлежащая рассматриваемой зоне локация (набор сот), в которой абонент находился некоторое отличное от 0 секунд время в течение рассматриваемого интервала. Если в течение рассматриваемого интервала найдутся несколько зон, для которых это условие выполнено, то абонент приписывается к каждой из них. Таким образом, отчет показывает количество уникальных абонентов, прошедшее через каждую зону в течение рассматриваемого периода.

2. Анализ требований и обзор входных данных

Требованиями к решению поставленной задачи являются:

– Предварительный расчет справочника по сетке административного деления г. Нижний Новгород по квадратам 500 на 500 метров за сентябрь 2015 года.

– Измерение zid должно соответствовать справочнику по сетке деления г. Нижний Новгород на квадраты 500 на 500 метров и содержать все зоны разбиения в результирующем отчете.

– Создание конечного программного продукта можно разделить на несколько взаимосвязанных этапов по разработке модулей формирования отдельных вспомогательных продуктов:

- справочник базовых станций;
- справочник покрытия базовых станций и их соседей;
- сетки административного деления 500 на 500.

Входными источниками являются:

- справочник с координатами полигонов для Нижнего Новгорода;
- версионный справочник по секторам свечения базовых станций по регионам;
- версионный справочник позиций базовых станций по регионам;
- справочник с сетками города. Размер ячейки в сетке – 500 на 500.

Наиболее важными при решении поставленных задач являются такие источники как:

– Трафик Anritsu. Содержит информацию о разобранных событиях сигнальной сети, включая следующие данные: звонки, смс, геолокацию БС. Представляет собой сжатые файлы в формате Sequence File для хранения в HDFS [1]. Источник содержит информацию по транзакциям в сетях 2G и 3G. Периодичность и время появления данных – каждые 15 минут. Средний объем поступающих данных в день – 55 Гб.

– Трафик Polystar. Похож по содержанию с Anritsu за исключением того, что включает в себя больше событий, больше регионов, а также ошибки сети. Средний объем поступающих данных в день – 12 Гб.

3. Формирование справочника уникальных базовых станций

Базовая станция (БС) – комплекс радиопередающей аппаратуры (ретрансляторы, приемопередатчики), осуществляющий связь с конечным абонентским устройством – сотовым телефоном.

LAC (Location Area Code) – код зоны местоположения группы базовых станций.

Cell ID – идентификатор конкретной соты внутри Location Area.

БС может быть описана при помощи комбинации LAC и Cell ID.

Для того чтобы рассчитать справочник БС, необходимо объединить файлы справочника по секторам свечения со справочником позиций базовых станций регионов по значению атрибута POSITION_CODE. В

результате получаем файл CITY_DIM_BASE_STATIONS.csv, поля которого, разделенные символом «;», приведены в таблице 1.

Таблица 1

Атрибуты выходных данных справочника БС

Показатель	Описание измерения
POSITION_LATITUDE	Географическая широта
POSITION_LONGITUDE	Географическая долгота
CELL	ID соты (номер вышки)
AZIMUTH	Направление свечения вышки
SECTOR_ANGLE	Угол свечения вышки
RANGE	Стандарт связи (900/1800/2100 МГц)

4. Алгоритм определения свечения вышки на ячейки сетки

На следующем этапе необходимо определить площадь свечения БС. Наносим координаты каждой БС на карту. Координаты БС (значения полей географических широты и долготы) берутся из справочника базовых станций. От получившейся точки откладываем прямую по направлению свечения БС (значение поля AZIMUTH из справочника CITY_DIM_BASE_STATIONS.csv). Затем от получившейся прямой откладываем угол, равный углу свечения вышки. В получившийся угол вписываем окружность с радиусом равным значению, соответствующему стандарту связи. Если для БС угол свечения строго больше 180 градусов, то около вышки описываем окружность радиусом 1 км, затем данную окружность вписываем в треугольник. Дальность свечения БС равна высоте треугольника. Если в справочнике по секторам свечения БС отсутствуют азимут и угол свечения, то вместо площади свечения будет использоваться круг. В дальнейшем получаем справочник покрытия с названием CITY_DIM_TRIANGLES_WKT.csv.

Формируем справочник по сетке 500 на 500. Используем заготовленную сетку по городу в виде .shp-файла – CITY_GRID_500.shp. На сетку из справочника города наносим покрытие БС в виде треугольников, получившееся в результате расчета CITY_DIM_TRIANGLES_WKT.csv. Если площади треугольников вышек пересекаются, то они являются соседними.

Далее определяем, на какие ячейки сетки светит данная БС. На рис. 1 приведен пример расположения БС и ее свечения на соседние квадраты сетки города:



Рис. 1. Расположение БС с разделением на квадраты

Рассчитываем площадь свечения вышки в каждой ячейке сетки (2, 3, 4, 6, 7, 8, 10, 11, 12) и заносим в справочник DIM_GRID_500.csv. Атрибуты справочника приведены в соответствии с содержимым таблицы 2:

Таблица 2

Атрибуты выходных данных справочника DIM_GRID_500.csv

Показатель	Описание показателя
ID_SQUARE	Идентификатор квадрата, на которую светит данная вышка
LAC	Номер группы вышек.
CELL_ID	ID соты (номер вышки).
AREA_GLARE	Площадь засвета квадрата, получившегося в результате пересечения квадрата и сектора вышки
AREA_COVERAGE	Площадь фигуры полученной в результате пересечения треугольника (покрытия БС) и квадрата (ячейки сетки)

5. Реализация модуля «Монобренд» и анализ полученного результата

Данный раздел описывает основные особенности непосредственной разработки и анализа полученных данных в результате эксперимента.

Разработку основного модуля «Монобренд» условно можно разделить на 3 этапа (по количеству stage-пакетов):

- фильтрация данных источников Anritsu и Polystar (пакет polyimsi);

- фильтрация данных справочника DIM_GRID_500.csv (пакет grid) и соединение данных, полученных на выходе первой стадии с отфильтрованными данными;
- группировка и подсчет значений требуемых показателей (пакет groupby).

Для первого этапа разработаны основные классы для получения событий сигнальной сети из источников – AnritsuMapper, PolystarMapper, PolyImsiReducer, реализующие абстрактные классы Mapper и Reducer из стандартного набора Hadoop-библиотек второй версии. Диаграмма классов представлена на рис. 2.

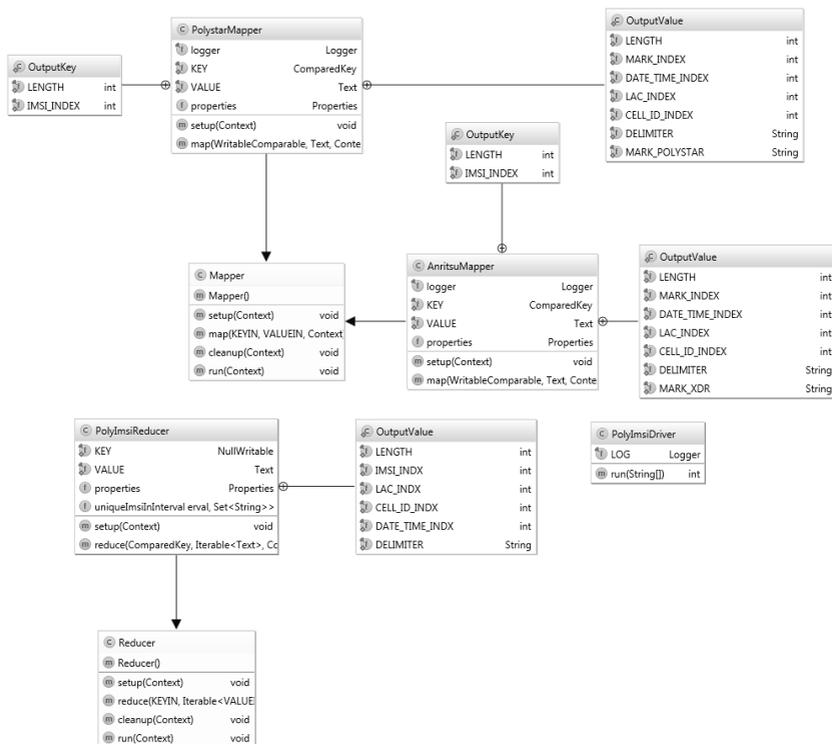


Рис. 2. Диаграмма классов пакета polyimsi модуля «Монобренд»

Для последующего анализа полученного результата потребовалось программное обеспечение QGIS Desktop 2.8.4 [2]. На основе имеющихся .shp-файлов строится карта города. Затем результирующий отчет привязывается к топологии по имеющимся координатам. Таким образом,

мы получаем нечто похожее на тепловую карту с указанием распределения полученных количественных показателей. Параметры цветовой гаммы настраиваются внутри программы QGIS по процентилям [2].

На представленном ниже рис. 3 отражена ситуация за 14 сентября 2015 года, а именно количество активных абонентов в промежуток времени с 9 утра до 21 вечера. Для обозначения точек продаж разных мобильных операторов, включая компании конкурентов, приняты условные обозначения. Они приведены как пояснения к рис. 3.



Фигуры на карте: «Билайн» – треугольники; «Мегафон» – квадраты; «МТС» – пятиугольники; «Теле2» – ромбы.

Рис. 3. Количество активных абонентов в городе Нижний Новгород в промежуток с 9 до 21 часов после полудня

Заключение

В ходе работы над задачей геоаналитики были разработаны все основные и вспомогательные модули, а именно:

- модуль формирования справочника базовых станций;
- модуль формирования справочника покрытия базовых станций и их соседей;
- модуль формирования сетки административного деления 500 на 500;
- модуль формирования результирующего отчета «Монобренд».

Результат работы программ, а именно полученный отчет, соответствуют критериям качества со стороны компании-заказчика, все

программные модули прошли тестирование и приняты как продуктивные задачи для автоматизированной работы на промышленном кластере.

Список литературы

1. White T. Hadoop: The Definitive Guide, 2nd Edition / T. White. – Sebastopol : O'Reilly Media, 2010. – 628 p.
2. Документация QGIS 2.0 [Электронный ресурс] : оф. сайт. – Режим доступа : <http://docs.qgis.org/2.0/ru/docs/index.html>

РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ СОЗДАНИЯ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ НА ОСНОВЕ ДЕФОРМИРУЮЩИХ ПРЕОБРАЗОВАНИЙ ИЗОБРАЖЕНИЯ

А.В. Швырева

Студент

А.А. Сирота

Профессор

Введение

В современном мире существует множество организаций, и каждая организация оперирует огромным количеством электронных и бумажных документов. В результате, возникают задачи проверки подлинности документов и защиты их от подделки. К методам, применимым для защиты документов, должны предъявляться следующие требования: они должны применяться без вмешательства третьей стороны (обеспечение конфиденциальности информации, минимальные временные затраты) и без использования специального оборудования (нет финансовых затрат). Один из наиболее эффективных методов защиты, отвечающий вышеперечисленным требованиям, заключается в использовании стеганографических методов защиты, а именно во встраивании в защищаемый объект цифровых водяных знаков (ЦВЗ). Для защиты электронных и бумажных документов может использоваться подход, основанный на внедрении ЦВЗ в специально создаваемые «дорожки» – изображения, состоящие из блоков, каждый из которых несет определенное количество информации от одного до нескольких бит.

1. Предлагаемый алгоритм создания водяных знаков для внедрения в цифровые изображения

Для защиты изображений применяются невидимые ЦВЗ [1, 2]. Как правило, внедрение ЦВЗ осуществляется в область исходного изображения

за счёт определённых преобразований яркости и цветовых составляющих изображения. В изображении ЦВЗ «рассеиваются» в младших битах цветности, что практически незаметно человеческому глазу, хотя при статистическом анализе они могут быть обнаружены. Для защиты от обнаружения используется «зашумление», приводящее статистические данные к нормальному виду (среднестатистическому) [2].

Предлагаемый алгоритм создания ЦВЗ в изображениях, используемых для защиты бумажных и электронных документов, подразумевает встраивание информации путем внесения деформирующих искажений (ДИ) в используемые для этой цели блоки изображения. Суть подхода заключается в создании ЦВЗ на основе деформации двумерных функций с помощью радиально-базисных функций (RBF). Использование интерполяции с применением RBF во многих случаях превосходит по точности другие виды интерполяции, поэтому в предлагаемом подходе будет использоваться интерполяция на основе RBF [3]. Математическая модель процесса внесения ДИ для исходной функции $f(x_1, \dots, x_n)$ n аргументов определяется следующим соотношением:

$$f(x_1 + r_1(x_1, \dots, x_n), \dots, x_n + r_n(x_1, \dots, x_n)) = g(x_1, \dots, x_n), \quad (1)$$

где $g(x_1, \dots, x_n)$ — результирующая деформированная функция; $r_i(x_1, \dots, x_n)$, $i = \overline{1, n}$ — функции вносимых деформирующих искажений по каждой координате.

Блок-схема предлагаемого алгоритма представлена на рис. 1:



Рис. 1. Блок-схема предлагаемого алгоритма внесения ЦВЗ

Для каждого блока входного изображения выполняется деформация на основе разложения по RBF. При внесении деформирующих искажений встраивается информация (битовая последовательность) и добавляется внешнее зашумление. Извлечение встроенных данных осуществляется с использованием однослойной нейронной сети с линейной функцией активации, реализующее двухальтернативное решающее правило.

2. Исследование алгоритма

Для исследования теоретических и практических вопросов создания ЦВЗ на основе деформации двумерных функций с помощью RBF было разработано приложение в среде разработки Matlab, реализующее вышеописанный метод.

В качестве входной функции использовалось изображение размером 50x50 пикселей. В это изображение встраивается 1 бит информации – последовательность $\{-1; 1\}$. В качестве деформирующей функции была выбрана квазидетерминированная функция, выполняющая перекося в угол:

$$f(x,y) = (\pm 1) * A * ok(x,y), \quad (2)$$

где A – амплитуда деформации; $ok(x,y)$ – оконная функция Гаусса, которая используется для сглаживания искажений, которые могут получиться в результате перекося. Использовалась RBF вида (3):

$$f(x) = \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right), \quad (3)$$

где x – вектор входных значений, m – центры RBF, σ – влияние RBF. Значение коэффициента зашумления задаётся в качестве константы и определяется в долях пикселя. Для извлечения информации создается нейронная сеть с одним нейроном (рис. 2).

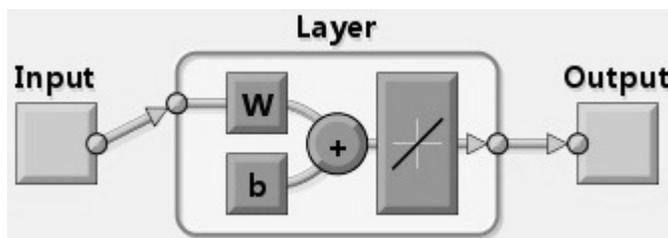
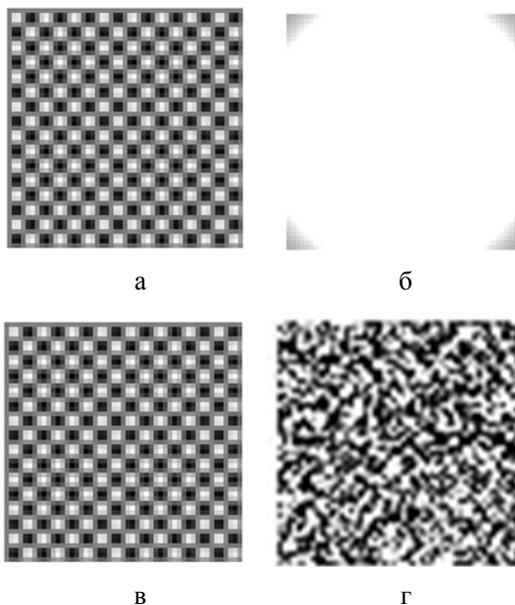


Рис. 2. Нейронная сеть, реализующая восстановление скрытой информации

После обучения и тестирования проводим оценку вероятности ошибки восстановления данных на основе обучающей выборки, сравнивая извлеченную последовательность с исходной.

В процессе тестирования программы были получены следующие результаты. Входное изображение представлено на рис. 3, а. На рис. 3, б представлен вид используемой деформируемой функции, на рис. 3, в вид деформированного изображения со встроенной информацией, а на рис. 3, г зашумленное изображение с уровнем шума 0.3 (в долях пикселя). Видно, что исходное и деформированное изображение неразличимы человеческим глазом.



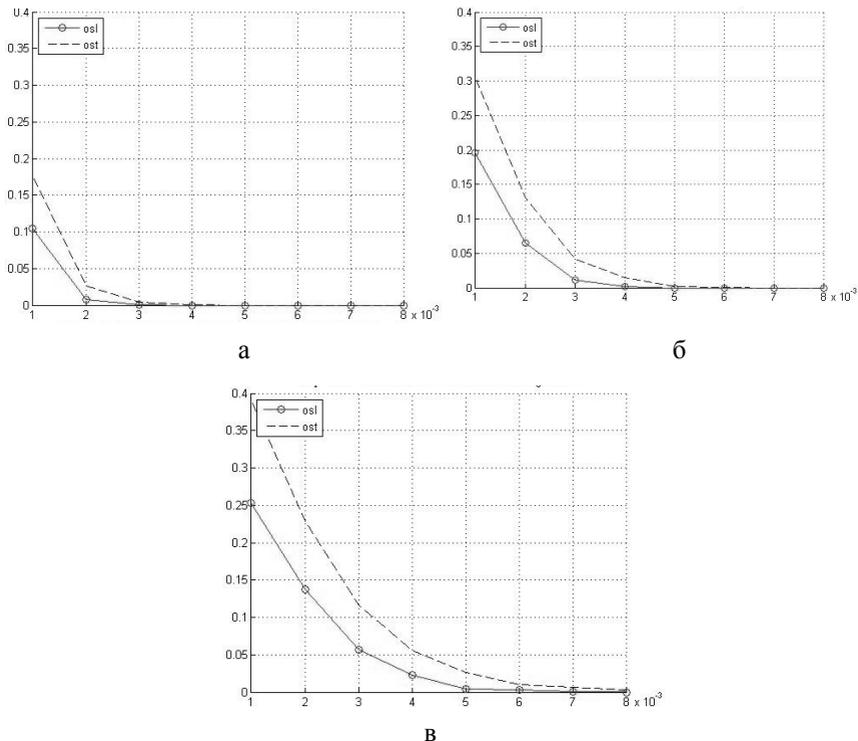
*а – вид исходного изображения; б – вид деформирующей функции;
в – вид деформированной функции; г – вид зашумленной деформированной функции*

Рис. 3. Результаты, полученные при тестировании

На рис. 4 представлены графики зависимости вероятности ошибки восстановления данных от амплитуды деформации в долях пикселя (при среднеквадратическом отклонении шума 0.3, 0.5, 0.7 соответственно). Пунктирной линией обозначена вероятность ошибки при восстановлении тестируемых данных, а сплошной – обучаемых. Видно, что нейронная сеть обеспечивает полное восстановление встроенных данных.

Анализируя эти графики, можно заметить, что с увеличением уровня шума вероятность ошибки восстановления данных увеличивается. Но, при

увеличении амплитуды деформации, эта ошибка стремится к нулю, из чего можно сделать вывод, что для полного достоверного извлечения информации необходимо комбинировать параметры уровня шума и амплитуды деформации.



а – зашумление = 0.3; б – зашумление = 0.5; в – зашумление = 0.7
 Рис. 4. Зависимость вероятности ошибки восстановления данных от амплитуды деформации в долях пикселя с разным уровнем шума

На рис. 5 представлен пример закрытого внешним шумом стегоконтейнера, состоящего из 8 блоков и содержащего последовательность {1, -1, 1, -1, 1, -1, 1, -1}.



Рис. 5. Заполненный стегоконтейнер

Заключение

В данной статье был предложен способ формирования цифрового водяного знака, способного нести в себе информацию, для физических и электронных документов. Для демонстрации предлагаемого способа было разработано приложение, реализующее встраивание последовательности $\{-1; 1\}$ в изображение и её извлечение с использованием нейронных сетей. Было проведено исследование предложенного алгоритма, в результате которого были определены зависимости вероятности ошибки восстановления данных от значения маскирующего шума.

Список литературы

1. Барсуков В.С. Еще раз о стенографии – самой современной из древнейших наук / В.С. Барсуков, А.В. Шувалов // Специальная техника. – 2004. – № 2. – С. 51-65.
2. Теренин А.А. Использование цифровых водяных знаков для борьбы с инсайдерами / А.А. Теренин, Ю.Н. Мельников, В.В. Погуляев // Специальная техника. – 2008. – № 1. – С. 27-30.
3. Акимов А.В. Модели и алгоритмы внесения деформирующих искажений на изображениях с использованием радиально-базисных функций / А.В. Акимов, М.А. Дрюченко, А.А. Сирота // Вестник ВГУ. Серия : системный анализ и информационные технологии. – 2013. – № 2. – С. 9-19.

СПОСОБ РАЗРАБОТКИ ТЕСТОВЫХ УДАЛЕННЫХ ИНФОРМАЦИОННО-ТЕХНИЧЕСКИХ ВОЗДЕЙСТВИЙ НА ПРОСТРАНСТВЕННО-РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ИНФОРМАЦИОННО-ТЕХНИЧЕСКИХ СРЕДСТВ

А.В. Щеглов

Студент

В.Ю. Храмов

Профессор

Введение

При разработке и эксплуатации пространственно-распределенных систем информационно-технических средств (ИТС) особое значение имеет оценка устойчивости их функционирования в условиях удаленных информационно-технических воздействий (ИТВ) со стороны злоумышленника. Несовершенство известных подходов к разработке таких

воздействий не позволяет анализировать реакцию информационно-технических средств на полное множество факторов, комплексно нарушающих точность и своевременность обрабатываемых ими сообщений. Эффективным путем решения данной проблемы является генерация тестовых информационно-технических воздействий на базе формализации процессов функционирования исследуемого объекта. Необходимо создание способа, обеспечивающего разработку полного множества тестовых удаленных информационно-технических воздействий на пространственно-распределенные системы информационно-технических средств на основе общедоступной спецификации их алгоритмов функционирования путем учета всех наборов взаимосвязанных факторов, комплексно нарушающих точность и своевременность обрабатываемых информационно-технических средств сообщений.

1. Анализ современных подходов разработки тестовых информационно-технических воздействий

С учетом ГОСТов [1, 2] в настоящей работе под ИТВ будем понимать последовательность действий по формированию совокупности факторов, нацеленной на нарушение конфиденциальности, целостности и (или) доступности обрабатываемой ИТС информации и (или) алгоритмов, и (или) электронных компонентов и (или) схем ее обработки при выполнении которого это средство переходит в состояние потери работоспособности, сниженной эффективности функционирования, управляемости или доступности для углубленного анализа источником воздействия.

На сегодняшний день для разработки тестовых ИТВ применяются следующие подходы:

1. эмпирический – основан на предположении аналитиками уязвимостей в ИТС и проверке этих предположений путем формирования и реализации соответствующего тестового ИТВ;

2. феноменологический – основан на накоплении фактов проявления уязвимостей в процессе функционирования ИТС и разработке соответствующих тестовых ИТВ, реализующих эти уязвимости;

3. использование аналогий – основан на анализе доступных баз данных с уязвимостями ИТС и реализующими их тестовыми ИТВ и выборе в этих базах данных подходящих тестовых ИТВ для исследуемой системы;

4. тестирование реализации – основан на разработке возможных тестовых ИТВ для текущей реализации ИТС, содержащих входные данные, сформированные по установленным правилам в целях обнаружения ошибок в этой реализации [3, 4];

5. проверка на модели – основан на представлении процесса функционирования ИТС в виде дискретно-событийной модели, в наборе состояний которой выделяется подмножество нештатных состояний;

6. генерация на основе модели – основан на таком стиле разработки ИТС, когда итерационно создается программная реализация его математической модели и автоматически генерируются программный код и тестовые ИТВ для проверки корректности кода.

Для решения указанной задачи первых двух подходов недостаточно, поскольку первый опирается на небезупречный человеческий фактор, а второй в течение длительного времени может не показать большей части уязвимостей. Третий подход ориентирован на поиск в ИТС известных уязвимых элементов. С его применением решить указанную задачу возможно только в том случае, когда ИТС состоит только из известных элементов. Четвертый подход сегодня достаточно глубоко проработан для алгоритмов функционирования программного и технического обеспечения ИТС с единственным потоком управления. Для случая распределенного многопоточного взаимодействия ИТС используется пятый подход. Однако его применение связано со сложностью определения состояний и формализации требований. Данные действия производятся эмпирически, и даже квалифицированный аналитик при работе с некоторым элементом локального алгоритма функционирования ИТС способен пропустить целый ряд уязвимостей. Шестой подход применим крайне редко, поскольку затраты только на его внедрение и создание программной реализации математической модели ИТС соизмеримы с затратами на разработку всего ИТС без модели. Поэтому на практике в подавляющем большинстве случаев для решения поставленной задачи реализуются комбинации первых четырех подходов, а адекватность их применения проверяется только временем. Такое положение дел для пространственно-распределенных систем ИТС не является удовлетворительным.

2. Содержание способа

В основе предлагаемого способа лежит модель функционирования пространственно-распределенной системы ИТС, базирующаяся на следующем положении: взаимодействие ИТС рассматривается с позиции обмена сообщениями, под которыми понимается как совокупность цифровых данных.

Наиболее полные определения локального и распределенного алгоритмов на сегодняшний день приведены в работе [5]. Однако в интересах достижения цели работы определение локального алгоритма требует дополнительного учета потоков управления и данных, а определение распределенного алгоритма требует следующих уточнений:

1) каждый локальный алгоритм в некотором состоянии распределенного алгоритма может находиться только в одном состоянии;

2) в структуре множества переходов распределенного алгоритма рассматриваются только те переходы локальных алгоритмов, которые связаны с событиями передачи и приема сообщений и тем самым способны оказать влияние на другие локальные алгоритмы.

Для определения элементов состояний распределенного алгоритма необходимо производить редукцию по правилу: каждая последовательность элементов локального алгоритма, начинающаяся после приема (передачи) сообщения и заканчивающаяся перед приемом (передачей) сообщения, заменяется новым состоянием.

Ниже приведен пример системы состоящих из трех ИТС (алгоритмы представлены уже в редуцированной форме), в первом ИТС используются два локальных алгоритма. В блоках и узлах схем и полученных из них графов показаны идентификаторы состояний локальных алгоритмов ИТС и приведены перечни соответствующих этим состояниям атомарных предикатов. Взаимодействие локальных алгоритмов в рассматриваемой системе и соответствующий распределенный алгоритм показаны на рис. 1.

Факторы удаленного ИТВ с позиции нарушения конфиденциальности, целостности и (или) доступности обрабатываемой ИТС информации способны влиять на характеристики сообщений:

Первая характеристика – своевременность, когда сообщения могут приниматься локальными алгоритмами в различных интервалах.

В одновременности, в течение которого приемное устройство все принятые сообщения рассматривает как одно. При раннем приеме алгоритм преждевременно переходит к последующим состояниям, в которых он отправит или будет ожидать поступления других сообщений. Принявший раннее сообщение локальный алгоритм будет выполняться с учетом данных или характеристик сигналов, заданных злоумышленником. Ранний прием может быть инициирован многократно в целях непредусмотренного расходования ресурсов ИТС. Поздний прием, равно как и невозможность приема, алгоритм воспримет как ошибку на физическом уровне;

Вторая характеристика – точность передаваемых в сообщениях данных или характеристик сигналов, которые согласно работам [3, 4] могут быть точными, больше, меньше граничных значений, равны им, а могут также быть неprecизионными (входят в интервал прецизионности) или неправильными (не в интервале прецизионности, но входят в допустимый численный интервал). Получение данных или характеристик сигналов за границами допустимого интервала алгоритм воспримет как сбой. Неprecизионные данные или характеристики сигналов незаметно

способны внести существенные погрешности в вычисления, выполняемые локальным алгоритмом и системой в целом, а неправильные способны дезориентировать локальный алгоритм и систему вплоть до применения ИТС во вред его системе. По аналогии с принятым в NASA подходом к тестированию бортового радиоэлектронного оборудования [6] в целях недопущения «взрыва» пространства тестируемых состояний при проверке одной цепочки в остальных входящих в цепочках принимаются точные и своевременные сообщения. Последовательная реализация каждой совокупности наборов, определяющей содержание группы тестовых удаленных ИТВ, позволит осуществить поиск ошибок в каждом состоянии алгоритма, связанном с обработкой сообщений от алгоритмов других ИТС.

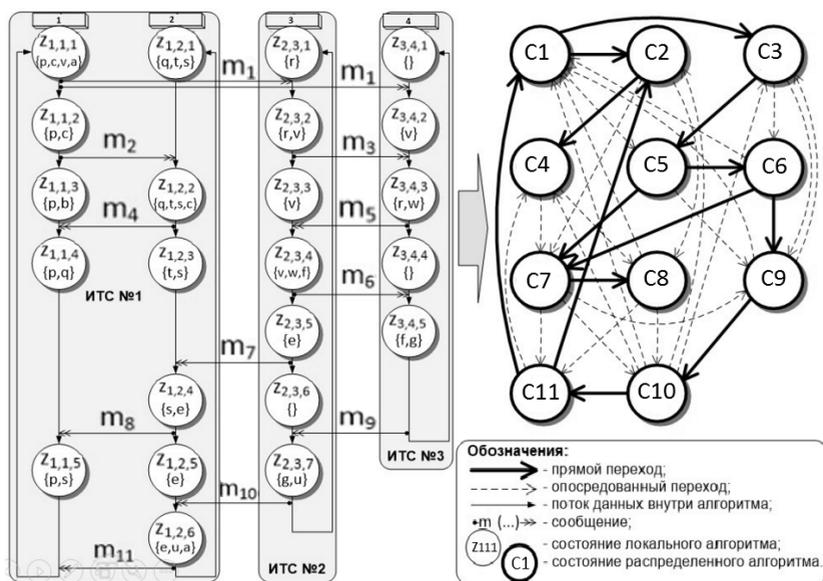


Рис. 1. Распределенная система ИТС и ее распределенный алгоритм

Предложенный метод разработки тестовых ИТВ состоит в следующем: выполняется построение множества локальных алгоритмов для всех ИТС системы; выполняется редукция локальных алгоритмов; выполняется построение распределенного алгоритма. Далее, фиксируется в каждом локальном алгоритме каждого ИТС множество состояний, непосредственно использующих неделимые элементы данных или характеристики сигналов, полученные от всех локальных алгоритмов других ИТС, элементы которого формируются правилом: определяется для каждого состояния, множество конечных цепочек из состояний локальных

алгоритмов. Цепочка описывает путь прохождения одинаковых данных или сигналов от состояния, в котором они возникли (получены), до состояния, в котором они используются. Определяется для каждого множества цепочек по одной группе тестовых удаленных ИТВ, включающей для каждой цепочки из множества факторы для всех возможных вариантов передачи сообщений (14 способов, основанных на различных комбинациях своевременности и точности).

3. Средство генерации тестовых информационно-технических воздействий на пространственно-распределенные информационно-технические средства

На основе предложенного метода было разработано программное средство, выполняющее генерацию полного множества тестовых информационно-технических воздействий на пространственно-распределенные информационно-технические средства.

Для создания пользователем модели пространственно-распределенной системы, было разработано приложение, написанное на языке #. Диаграмма классов данного приложения изображена на рис. 2.

Для генерации тестовых информационно-технических воздействий на предложенной пользователем модель информационно-распределенной системы был разработан программный модуль на языке c++. Программы, написанные на двух разных языках, обмениваются с помощью объектных файлов XML. В данном файле содержится информация о состояниях локальных алгоритмов: номер шага алгоритма, используемые предикаты, номера входящих или исходящих сообщений, номер алгоритма и ИТС.

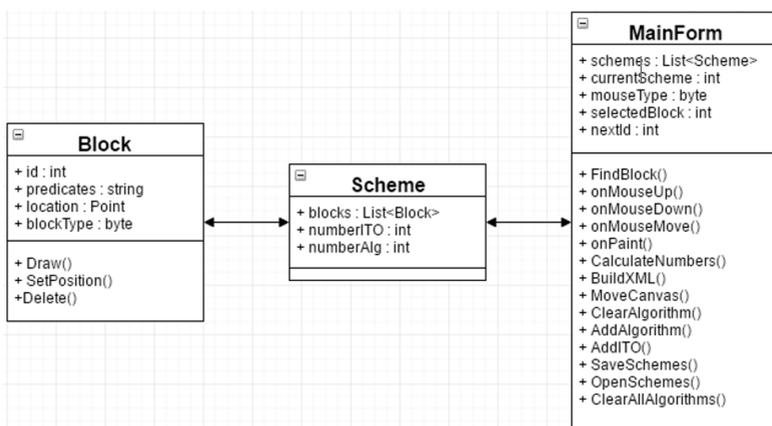


Рис. 2. Диаграмма классов приложения создания блок-схем локальных алгоритмов распределенных ИТС

Вывод части результата для предложенной модели в данной статье представлены в таблице.

Таблица

Вывод множества информационно-технических воздействий на предложенное пользователем пространственно-распределенное информационно-техническое средство

Состояния	Предикаты	Цепочки к состоянию	Временные характеристики	Кол-во тестов
$z_{1,2,6}$	e	$z_{2,3,5} \# z_{1,2,4} \# z_{1,2,5} \# z_{1,2,6}$	$\Delta t_0 = [t(m_{11}), t(m_{11}) + \Delta t_{\min}]$; $\Delta t_{>} = [t(m_6) \vee t(m_8) \vee t(m_9) \vee t(m_{10}), t(m_{11})]$; $\Delta t_{<} = [t(m_{11}), t(m_{11}) + \Delta t_{\text{wait}}]$	28
	u	$z_{2,3,7} \# z_{1,2,6}$	$\Delta t_0 = [t(m_9), t(m_9) + \Delta t_{\min}]$; $\Delta t_{>} = [t(m_7) \vee t(m_8), t(m_9)]$; $\Delta t_{<} = [t(m_9), t(m_9) + \Delta t_{\text{wait}}]$	

Функциональность программного средства включает в себя следующие возможности:

- Создавать и редактировать блок-схемы (создание, перемещение, удаление блоков, редактирование предикатов), соединять созданные блок-схемы;
- Добавлять /Удалять алгоритмы (текущего или другого ИТС);
- Загрузка/Сохранение в файл/из файла всех добавленных алгоритмов (данная возможность получена благодаря сериализации языка C#);
- Генерировать тестовые информационно-технической воздействий на предлагаемую пользователем модель распределенной информационно-технической системы.

В результате работы программного средства генерируется HTML-файл, содержащий полный набор тестовых информационно-технических воздействий (в виде таблицы) на предложенную пользователем модель распределенного информационно-технического средства. Данный файл автоматически открывается браузером по умолчанию.

Заключение

Таким образом, разработано программное средство, позволяющее генерировать полное множество информационно-технических воздействий на предложенные пользователем пространственно-распределенные информационно-технические средства, представленным в статье способом с использованием понятий локального и распределенного алгоритмов,

формируемых на основе общедоступной спецификации об алгоритмах функционирования информационно-технических средств пространственно-распределенных систем. Данный способ впервые позволяет обеспечить разработку полного множества тестовых удаленных информационно-технических воздействий путем учета всех потенциально возможных наборов факторов, комплексно нарушающих точность и своевременность обрабатываемых данными средствами сообщений.

Список литературы

1. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения [Текст]. – Введ. 2006-12-27. – М. : Изд-во стандартов, 2007. – 8 с.
2. ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения [Текст]. – Введ. 2006-12-27. – М. : Изд-во стандартов, 2007. – 8 с.
3. Myers G. J. The Art of Software Testing / G. J. Myers. – New Jersey : John Wiley & Sons, 2004. – 255 p.
4. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер. – СПб. : Питер, 2004. – 318 с.
5. Tel G. Introduction to Distributed Algorithms / G. Tel. – N. Y. : Cambridge University Press, 2001. – 612 p.
6. Hayhurst K. J. A Practical Tutorial on Modified Condition. Decision Coverage / K. J. Hayhurst, D. S. Veerhusen, J. J. Chilenski. – Hampton : Langley Research Center, 2001. – 85 p.

USING DATA MINING METHODS TO OPTIMIZE OVERBOOKING

Osama Alkaadi

Student

Mothana Khoder

Student

S.V. Vlasov

Associate Professor

Introduction

This paper deals with the use of advanced statistics and data mining techniques to extract knowledge from large databases containing booking information in online reservation systems (airlines, trains, hotels, hospitals, ...). This knowledge will be used to predict subscriber's behavior, which in turn is used to optimize capacity planning and improve overbooking management. Overbooking is a common way in the travel and residence industry, in which it is expected that some people will cancel. Number of those people called "No-Show". Reservation companies resort to increase the number of bookings in order to get good financial revenues in case of many cancellations. Optimization of overbooking very important: in case of overestimation of the actual "no-show", bookings for some customers will be denied and have to receive some compensation, which represents a cost for the reservation company, together with the associated loss in customer satisfaction; on the other hand, underestimated "no-show" may cause losses of potential revenues.

In paper will describe the currently used methods for predicting "no-show" probability of customers then talk about using new data mining methods [1]. These methods include Decision Trees, Logit Models and Neural Networks.

1. The Current Methods to Produce Predictions

Currently, the weighted mean method (WM) [2] is used to produce the "no-show" predictions. The calculation is based on the mean of the historical "show-up" rate, weighted by the number of final bookings; the show up rate (SUR) is simply the ratio between the number of final shows (so called booked out), and the number of final bookings [3]. After that, MAD (Mean Absolute Deviation) [4], which is an improvement of the various models over the original Weighted Mean method used. MAD method has several advantages: it is very simple and it relies on little data which are always available at any time. But MAD typically exhibits a high prediction error. The wrong predictions of the MAD method correspond in the best case to an error rate of around 30% [5].

2. Using Data Mining Methods and Techniques

Knowledge management and data mining techniques have been widely used in many important applications. Considering that accurate forecasts are the base for correct decisions in revenue management, the target of this paper is to study the role of knowledge management and data mining techniques for a demand estimating process. Critical success factors are: (a) collection and preparation of data; (b) the method used for data reduction; (c) the forecasting methods. Data collection and preparation is essential to clean and carefully check the available data coming from different sources. I am going to use Decision trees method[1] for exploratory analysis and data reduction. Decision trees are a very practical way of sifting through data and finding relationships and patterns. They are important as an analytical modeling procedure but can also be used to identify likely variables for inclusion in the modeling procedure. Decision tree analysis will always produce a result even if there are only weak relationships between the target and input variables. If it is likely that the relationships are weak and also if there is not much data available, then it can be helpful to construct a random forest. The application of the CHAID method [6] will be used in this project on the ground that CHAID is a type of decision tree technique. Logit model [7], which is a linear regression model well-suited for conditional probability estimation. The preference is for decision tree analysis or logistic regression. In either case, the target is a binary variable with just two values such as 0/1. The input variables do not need to be Normal for decision trees; Normality is more important for logistic regression. Transformations help to make the input variables suitable. Note that the model is likely to need updating after the next stages of evaluation and validation. Neural networks (also called neural nets) are the data mining method attracting the highest attention, but which actually have the smallest number of operational applications. The application of neural networks to data mining comes from the area of ‘machine learning’. Therefore, this study will use back propagation neural networks (BPN) and general regression neural networks (GRNN) [8] to construct customer cancellation prediction models, assisting managers in making customer cancellation judgments and improving their prediction accuracy to avoid unnecessary loss regarding costs.

3. Artificial Neural Networks

This study will use Back Propagation Neural Network (BRN) and General Regression Neural Networks (GRNN) to build a customer cancellation prediction model and use 14000 customer reservation data taken from Syrian Airlines database. I will use 9 customer attribute and consumer record variables will be taken into consideration : day, month, year, age, sex, marital status, education, history of cancellation, and last number of cancellations.

In this case the BRN and GRNN Networks have 9 input layer neurons and one neuron in the output layer (the output values are 0 or 1, 0 means completed reservations, 1 means cancellation. This study will analyze the prediction model's performance and identification abilities according to specificity, sensitivity, and receiver operating characteristics (ROC) [9]. In statistics, a receiver operating characteristic (ROC), or ROC curve, is an effective method of evaluating the performance of diagnostic tests. The curve is defined as a plot of true positive rate (TPR=TP/P) or sensitivity as the y coordinate versus the false positive rate (FPR=FP/N) or (1-specificity) as the x coordinate for different cut-off points of a parameter. Table 1 shows how we can calculate TPR and FPR.

Table 1

Confusion Matrix (Contingency Table)

Predicted class	TRUE CLASS	
	p (positive)	n (negative)
Yes	True Positives	False Positives
No	False Negatives	True Negatives
Total	P	N

Larger spaces under the ROC Line represent the ability of prediction model to determine the data type, smaller areas represent lower identification ability. When the BPN and GRNN prediction models completed their learning, we used 4200 sets of data to test the models' accuracies. Test results in Tables 2 and 3.

Table 2

BPN test results

1: signified cancellation / 0: signified fulfilled appointments

	Actual Value 1	Actual Value 0	Total
Prediction Value1	560	880	1440
Prediction Value0	140	2620	2760
Total	700	3500	4200

Table 3

GRNN test results

1: signified cancellation / 0: signified fulfilled appointments

	Actual Value 1	Actual Value 0	Total
Prediction Value1	610	1070	1680
Prediction Value0	90	2430	2520
Total	700	3500	4200

In addition to the visual component, there is a numerical characteristic of ROC : AUC the area under the ROC curve, the higher the value the better. Specificity, Sensitivity, and the square under the ROC curve shown in Table 4 which indicates that Both models produced acceptable results for classification. The ROC is shown in Figure, which indicated that BPN had greater classification accuracy results than did GRNN.

Table 4

Test results of BPN and GRNN

	Sensitivity	Specificity	Square under curve
BPN	80%	75%	80.87%
GRNN	87.5%	69.5%	75.34%

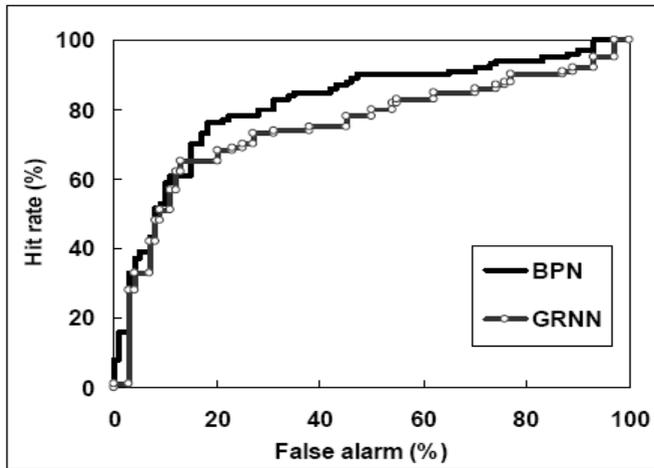


Figure. ROC curves for both ANN models.

Conclusions

Different companies depend on managers' experience to guess the number of cancellations, and the resulting prediction errors can result in surpluses or insufficient service capacity. This project is going to use neural networks to build cancellation prediction models. I am going to confirm that both prediction models have excellent cancellation predictive abilities, can help managers in judging whether customers will cancel reservations, and can help in dynamic service capacity scheduling.

REFERENCES

1. Shirley C. A Practical Guide to Data Mining for Business and Industry / C. Shirley, S. Andera; SPi Publisher Services. – 1st edition. – Pondicherry : Minion, 2014. – 303 p.
2. Mark G. [and others.] GNU Scientific Library / G. Mark; Free Software Foundation. – Edition 2.1. – Boston : Invariant, 2015. – 589 p.
3. Janakiram S. Airline Yield Management with Overbooking, Cancellations, and No-Shows / S. Janakiram, S. Shaler, L. Conrad // Journal Transportation Science. – Maryland, USA, 1999. – Vol. 33. – Issue 2. – P. 147-167.
4. Elsayed A. Mean Absolute Deviation About Median As A Tool Of Explanatory Data Analysis / A. Elsayed // International Association of Engineers : collection of the materials of scientific conference. “Proceedings of the World Congress on Engineering 2012 Vol. I” (London, July 4-6, 2012). – London, 2012. – P. 324-329.
5. Francesco V. An Application of Data Mining Methods to Airline Overbooking Optimization / V. Francesco // International Fuzzy Systems Associations (IFSA) : collection of the materials of scientific conference. “Fifth International Conference on Application of Fuzzy Systems and Soft Computing” (Milan, September 17-18, 2002). – Milan, 2002. – P. 88-93.
6. Antipov E. Applying CHAID for logistic regression diagnostics and classification accuracy improvement / E. Antipov, E. Pokryshevskaya // Journal of Targeting, Measurement and Analysis for Marketing. – London, UK, 2010. – Vol. 18. – P. 109-117.
7. Hilbe M. Practical Guide to Logistic Regression / M. Hilbe; Taylor & Francis Group. – 1st edition. – London : CRC Press, 2015. – 174 p.
8. Merry C. Neural Network Based ACC for Optimized Safety and Comfort / C. Merry, S. Paul // International Journal of Computer Applications. – Pennsylvania, USA, 2012. – Vol. 4. – P. 1-4.
9. Wojtek J. ROC Curves for Continuous Data / J. Wojtek, J. David; by Chapman and Hall / CRC. – London : CRC Press, 2009. – 256 p.

IMPROVED HEURISTIC LOAD BALANCING ALGORITHM

Mothana Khoder

Student

Osama Alkaadi

Student

S.V. Vlasov

Associate Professor

Introduction

Grid computing is a kind of distributed computing that involves both the integrated and the collaborative use of distributed resources in order to finalize huge amounts of computational tasks which require reliable resources sharing across computing domains . Load balancing is very important in this way of computing as it can improve the performance of the grid computing . This paper attempts to develop an algorithm for the heuristic load balancing depending one of these dynamical algorithms which is well known as the Ant Colony Optimization (ACO) approach . We especially concentrate on three issues which have not been discussed before: The first issue is that in this work we take into account the existence of local load in the network resources this parameter has not been discussed before. The second issue is that we will assign different costs for the different tasks depending on the required resources to finalize the task. This new feature in the study expands the possible options for the customers. The third issue is the possibility for moving the some of the tasks between the different resources according to their availability in such a way we move some tasks from the overloaded resource to the resource with less tasks in dynamical way during the real executing time.

The proposed algorithm is implemented with GridSim toolkit which can simulate a decentralized module. The GridSim toolkit abstracts the features and behavior of complex fundamental grid elements such as grid tasks, grid resources and grid users . For evaluation purpose, a comparison of execution times and cost of proposed algorithm and the other similar algorithm is also provided in this paper.

1. Ant Colony Optimization

ACO is a meta-heuristic using artificial ant to find desirable solutions to difficult combinatorial optimization problems [1, 2]. The behavior of artificial ants is based on the traits of real ants plus additional capabilities that make them more effective, such as a memory of past actions. Each ant of the "colony" builds a solution to the problem under consideration and uses information

collected on the problem characteristics and its own performance to change how other ants see the problem. Ants communicate with each other through the use of pheromones. As ants traverse a path, they deposit pheromones [3]. Pheromones are chemical substances that attract other ants and are deposited by ants on the ground as they travel. Ants move randomly, but when they encounter a pheromone trail, they decide whether or not to follow it. If they do so, they lay down their own pheromone on the trail as well, reinforcing the pathway. The probability that an ant chooses one path over other increases proportional to the amount of pheromone present. The more ants that use a given trail, the more attractive that trail becomes to subsequent ants [4, 5].

2. Improved Heuristic Load Balancing Algorithm (IHLBA)

An algorithm for load balancing is successful, if the task executes within the deadline. All the tasks in this algorithm are distributed according to Ant Colony Algorithm. In IHLBA [6, 7]. Let the number of tasks (ants) in task set T maintained by task agent is P and the number of registered resources is K. When a new resource R_i is registered to GIS, then it will initialize its pheromone based on:

$$PH_i^{init} = [(N \times M) \times (1 - load_{local})] / RC_i, \quad (1)$$

where N represent the number of processing elements and M corresponds to MIPS rating of processing element. $load_{local}$ is the local load in the resource . And RC_i represents the cost of using resource in the time unit. When the user sends his tasks to the grid network to be performed, each resource gives the pheromone a value using the following equation:

$$PH_i^{t=0} = \left[\left(\frac{PH_i^{init}}{PH_{min}} \right) \times AL \right] / \sum_{j=0}^k PH_j^{init} \quad (2)$$

in which AL represents the sum of all the lengths of the tasks according to MIPS. The task is assigned to the resource with the larger pheromone value then we decrease the value of the pheromone by the C which represents the complication of the computing for the task (the length of the task in MI)

$$PH_i^t = PH_i^t - C, \quad (3)$$

when the task is done we increase the value of the pheromone in the same amount.

$$PH_i^t = PH_i^t + C. \quad (4)$$

Next, the new value of the pheromone R_i is compared with the value ($Average + \sigma$) where *Average* is the average of the pheromones values for all the resources and σ is the standard deviation of this values. Next, we search for resources that the value of their own pheromone less than ($Average - \sigma$) and we

transfer the tasks from these resources to R_i . The notations used in the description of IHLBA are illustrated in Table 1.

Table 1

Notations Used

T	Number of Tasks
K	Number of resources
R_i	i^{th} Available Resource
N	Number of processing elements in R_i
M	Rating of processing element corresponds MIPS
load _{local}	Local load in resource
RC_i	Cost of using resources
AL	Total lengths for all tasks corresponds to MIPS
C	The length of a single task corresponds to MIPS
ET_i	Time resource usage
PH_i^{init}	Initial Pheromone Associated with R_i
PH_i^t	Current Pheromone Associated with R_i
FT_i^t	Time execution of a single task at resource
AJN	Assigned Job Number
Average	Average values pheromones of resources
σ	The standard deviation of values pheromones
GridLoad _{i}	Grid load in resource R_i

The working of proposed algorithm is as follow:

Step 1. Initialize the value of T, K, N, M, load_{local}, RC_i , AL, C, ET_i .

Step 2. Calculate the initial value of the pheromone for each resource at the registration time at the GIS.

$$PH_i^{init} = [(N \times M) \times (1 - \text{load}_{local})] / RC_i .$$

Step 3. Calculate the value of the pheromone for each resource at the beginning of the work.

$$PH_i^{t=0} = \left[\left(\frac{PH_i^{init}}{PH_{min}} \right) \times AL \right] / \sum_{j=0}^k PH_j^{init}$$

Step 4. Select the next task t.

Step 5. Find resource R_i with high value pheromone among all resources

$$PH_i^t = \max (PH_j^t) : j \in K ,$$

i. e. resource R_i is having minimum load.

Step 6. Set $T = T - 1$.

Step 7. Assign task t to R_i .

Step 8. decrease pheromone of R_i as:

$$PH_i^t = PH_i^t - C.$$

i. e. reporting R_i as heavily loaded.

Step 9. Check whether any task failure or completion. If no, go to Step 13.

Step 10. If (task failure at any resource R_i) then add task again to the tasks list.

Step 11. If (task completion at any resource R_i) then Increase pheromone of R_i :

$$PH_i^t = PH_i^t + C$$

i. e. reporting R_i as lightly loaded

$ET_i = ET_i + FT_i'$: to identify usage time of each resource;

$ATN = ATN + 1$: to identify the number of task which have been performed by the resource.

Step 12. If $PH_i^t > (\text{Average} + \sigma)$.

Search for the resource which its pheromone is smaller than $(\text{Average} - \sigma)$ and remove one of its tasks, and, go to Step 7

Step 13. If $(T > 0)$ then go to Step 4.

Step 14. For each resource R_i , $0 \leq i \leq K$ compute:

$$\text{Execution Cost} = \sum_{i=1}^K ET_i * RC_i,$$

$$\text{GridLoad}_i = \frac{AJI}{\sum_{j=1}^N M_j}.$$

3. Simulation Results

In this section, we discuss the results of simulations of the proposed our algorithm based on a set of standards then we compare its performance with (HLBA) algorithm, and random algorithm (RA) [8], which randomly select resources from a list of available resources. The table 2 shows the characteristics of the resources used in the simulations.

Table 3 shows the characteristics of the tasks used in the completion of the simulation, assuming that these tasks independently and don't require a particular order for implementation.

Table 2

Characteristics of the resources

Parameter	Value
Number of resources	50
Local load in resource	0.2 -0.8
Number of processing elements in resource	1-4
Rating of processing element corresponds to MIPS	10 or 50 MIPS
Cost of using resources	0.1 to 0.5 G/\$
Bandwidth	50 – 150 b/s

Table 3

Characteristics of the tasks

Parameter	Value
Number of Tasks	50-200
Length of the task (MI)	100-20000
Size input file	100+(10% to 40%)
Size output file	100+(10% to 50%)

The following criteria are used in evaluating the performance of algorithm[7, 9]:

- The cost of task's execution, which represents the total cost of execution of the tasks of the application which are allocated to the available resources.
- The total time which is needed to execute the application Makespan. This will be measured from the time of sending the first task to the time of reception of the last task of the application.
- The level of the load balancing in the system .It is a criterion to test the performance of the load balancing algorithm. If the algorithm is efficient the load balancing value will be close to 100% while the standard deviation value is close to zero this criterion is given by the following equation:

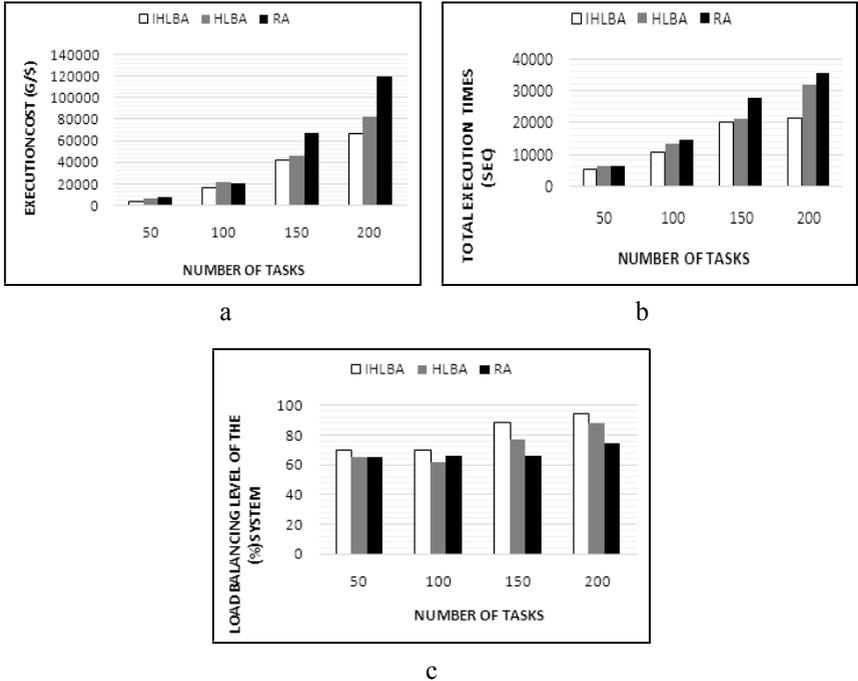
$$\text{LoadBalancingLevel} = (1 - d) * 100\% . \quad (5)$$

To perform the simulations, we create a user which generates tasks. These tasks are independent from each other (task characteristics are described in Table 3). The user sends the tasks to a Grid network that contains a fixed number of processing resources (resource properties are described in Table 2). We divide the simulation into four stages in every stage a different number of tasks is sent by the user . The simulation results are shown in the following figures:

Figure, a shows the cost of task's execution for the different algorithms according to the number of the executed tasks. Figure, b shows the total time to

execute the application for the different algorithms according to the number of the executed task

Figure, c shows the level of the load balancing in the system for the different algorithms according to the number of the executed task.



a – cost of task's execution, b – total time to execute the application, c – the level of the load balancing in the system

Figure. Comparison of algorithms

4. Results Discussion

As can be seen in the figure, it is clear that the suggested algorithm is better than the other algorithms. This can be understood due to its ability to take into account the local load issue and the cost of the task's execution when the tasks are allocated to different resources.

The other advantages of the suggested algorithm are:

- The fast response to the events which can cause a change in the load the possibility of reallocate the load in different way when unbalanced load is discovered

– The possibility of moving the tasks from the overloaded resources to the less loaded resources.

In the table 4, we show the average percentage ratio of the improving in application execution using our algorithm in comparison with the other algorithms.

Table 4

Average percentage ratio of the improving

Percentage of improving	Load Balancing Level of the System	Total Execution times	Execution Cost
Compared with the HLBA	8%	21%	17%
Compared with the RA	13%	33%	40%

CONCLUSION

In this work, we proposed a new algorithm to improve the heuristic load balancing for grid computing. We improved the load balancing using the ant colony dynamical approach by taking the local load in the network. We optimize the usage of our resources by transferring some tasks between different resources according to their availability. The results show the suggested algorithm is better in load allocation to the resources than the other algorithms. The time and the cost using the suggested algorithm are less than the other algorithms this allows this algorithm is a good choice to be used. The ability of the algorithm to update itself with the events which can change the load status allows it to be a good choice to be used in the grid computing environment. These results show the improvement in the load balancing using our approach in comparison with the other approaches.

REFERENCES

1. Rohil H. A Heuristic Based Load Balancing Algorithm / H. Rohil, S. Kalyan // IJCEM International Journal of Computational Engineering & Management. – Newdelhi, 2012. – Vol. 15. – P. 56-61.
2. Maozhen L. The Grid Core Technologies / L. Maozhen, B. Mark; SPi Publisher Services. – 1st edition. – Berlin : Pondicherry, 2005. – 452 p.
3. Rajkumar B. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing / B. Rajkumar, M. Manzur; by Chapman and Hall / CRC. – London : CRC Press, 2009. – 256 p.
4. Mohammad H. Efficient Load Balancing Using Ant Colony Optimization / H. Mohammad, S. Nadimi // Journal of Theoretical and Applied Information Technology. – 2015. – Vol. 23. – P. 253-258.

5. Chellamani R. A Decentralized Dynamic Load Balancing for Computational Grid Environment / R. Chellamani, R. Sivaranjani // Journal on Soft Computing. – 2011. – Vol. 3. – P 610-614.
6. Bansal B. Design and development of grid portals / B. Bansal, B. Seema; by Chapman and Hall / CRC. – London : CRC Press, 2012. – 438 p.
7. GNU Scientific Library / G. Mark [and others]; free Software Foundation. – Edition 2.1. – Boston : Invariant, 2015. – 589 p.
8. Schopf J. Grids: the top ten questions. Scientific programming / J. Schopf; free Software Foundation. – Edition 2.1. – Boston : Invar, 2015. – 589 p.
9. Foster I. What is the grid? a three point checklist / I. Foster // Grid Today. – 2013. – Vol. 6. – P. 22-25.

**РАБОТЫ, ВЫПОЛНЕННЫЕ СОВМЕСТНО С
ВОРОНЕЖСКИМ ГОСУДАРСТВЕННЫМ МЕДИЦИНСКИМ
УНИВЕРСИТЕТОМ ИМ. Н.Н. БУРДЕНКО**

**ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ДИАГНОСТИКИ
ИНФЕКЦИОННЫХ ЗАБОЛЕВАНИЙ**

Г.В. Измайлов
Студент
Т.Ю. Брусенская
Аспирант
П.В. Саенко
Ординатор
С.В. Борзунов
Доцент
С.Д. Кургалин
Профессор
Ю.Г. Притулина
Профессор

Введение

В настоящее время в Воронеже в области диагностики и лечения инфекционных заболеваний печени специализируются пять государственных медицинских учреждений [1]:

- Областной клинический центр профилактики и борьбы со СПИД;
- Областная клиническая инфекционная больница;
- Городская клиническая поликлиника №4;
- Городская поликлиника №7;
- Городская поликлиника №8.

К сожалению, данные учреждения не имеют единой базы данных пациентов. Это вызывает у врачей значительные трудности: обмен информацией занимает немалое время, возможна потеря данных, истории болезни пациентов, хранящиеся в одном учреждении, оказываются недоступными в другом.

© Измайлов Г.В., Брусенская Т.Ю. (ВГМУ им. Бурденко),
Саенко П.В. (ВГМУ им. Бурденко), Борзунов С.В., Кургалин С.Д.,
Притулина Ю.Г. (ВГМУ им. Бурденко), 2016

В силу этого представляется актуальной задача построения единой базы данных для этих учреждений [2].

1. Требования к программному комплексу

Цель работы заключалась в разработке программного комплекса для учреждений, специализирующихся в области диагностики и лечения инфекционных заболеваний печени [3].

К разрабатываемому программному комплексу были выдвинуты следующие требования:

- использование базы данных, в которой хранились бы все необходимые данные о пациенте;
- интегрируемость базы данных для нескольких профильных учреждений, где решаются различные задачи и вносятся данные разных типов;
- наличие модуля с обучающей частью, т. е. своего рода простейшей экспертной системы, помогающей молодому специалисту принять верное решение при постановке диагноза.

Требования к составу и параметрам технических средств обусловлены тем, что комплекс будет использоваться в различных учреждениях здравоохранения. Состав технических средств определен следующим образом:

- интернет-сервер для хранения базы данных;
- средство для создания базы данных – MySQL Workbench 6.0;
- система управления базой данных MySQL;
- язык программирования C#.

2. Реализация базы данных

База данных была создана в приложении MS Workbench. Её особенность заключается в том, что она ориентирована на пациента, что, в свою очередь, позволяет отследить динамику его заболевания [4]. Все таблицы в базе данных можно разделить на четыре группы:

- таблицы для заполнения информацией о месте обследования и специалисте, проводящим его;
- таблицы для хранения результатов анализов;
- таблицы данных, получаемых в результате осмотра врачей;
- таблицы, хранящие основные сведения о пациенте.

На рис. 1 представлена схема базы данных, используемой в приложении.

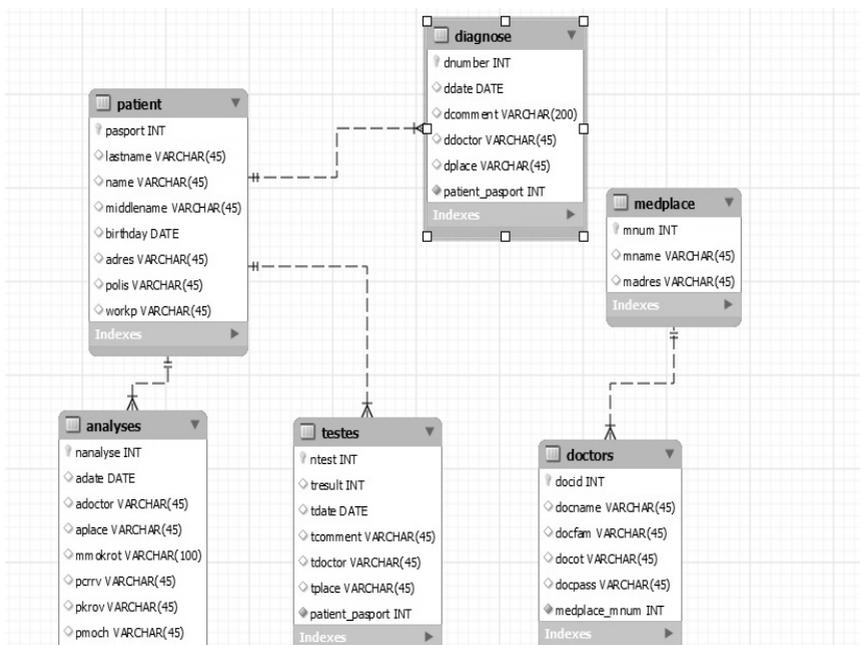


Рис. 1. Схема базы данных программного комплекса для диагностики инфекционных заболеваний

3. Реализация интерфейса приложения и обучающей части

На главной форме приложения предлагается найти нужного пациента в имеющейся базе данных. Если такого пациента нет, то имеется возможность добавить новую запись об этом пациенте. Сделать это можно после того, как заполнены все обязательные поля, например: паспортные данные, фамилия, имя, отчество, номер полиса, дата рождения, место работы, адрес. При вводе в поля формы некорректных значений выдается сообщение об ошибке. Также имеется возможность редактирования существующих данных. Например, пациент достиг определенного возраста и ему нужно менять паспорт. Для сохранения его истории болезни необходимо изменить поле «паспортные данные», оставив всё остальное без изменения. Программа предусматривает эту возможность. Предположим, нужный пациент найден в базе, тогда следует изменить данные, относящиеся к паспорту. Для этого устанавливается отметка о том, что мы хотим изменить сведения и произвести это действие.

Когда данные корректны и нужно продолжить работу с выбранным пациентом, по «щелчку» по его данным открывается новая форма, которая предназначена для заполнения сведений об осмотре. В этой форме врач должен ввести данные о себе, указать время обследования, выбрать название учреждения и указать диагноз пациента.

В каждом учреждении перечень анализов, возможных для сдачи, разный, поэтому в форме для внесения данных об анализах не все поля обязательны для заполнения. Доктор вводит результаты обследований, и они сохраняются в базе данных и становятся доступными для последующего просмотра.

Последняя форма – это простейшая обучающая система, призванная помочь молодому специалисту принять правильные решения. В этой форме находится множество обязательных для заполнения полей, соответствующих названиям симптомов, которые имеются у больного при обращении к врачу. Группа симптомов формирует синдромы, которым присвоена различная степень важности. На основе программа делает заключение о состоянии здоровья пациента. Реализация алгоритма расчета вероятности заболевания состоит в вычислении суммы слагаемых с различными весами, причем их веса присваиваются экспертом в предметной области.

4. Результаты работы программного комплекса

Программный комплекс осуществляет структурирование важной информации, просмотр всех данных о больном возможен с помощью нескольких нажатий кнопок. Врачу будут доступны все данные о пациенте, его полная история болезни. Реализована электронная карточка для каждого больного. Обучающая система с большой долей вероятности помогает принять правильное решение.

Результат работы программного комплекса представлен на рис. 2 и рис. 3. Видно, что интерфейс комплекса является развитым и, безусловно, позволяет врачу-диагносту проанализировать большое количество необходимой ему информации, чтобы сделать правильный вывод о состоянии здоровья пациента.

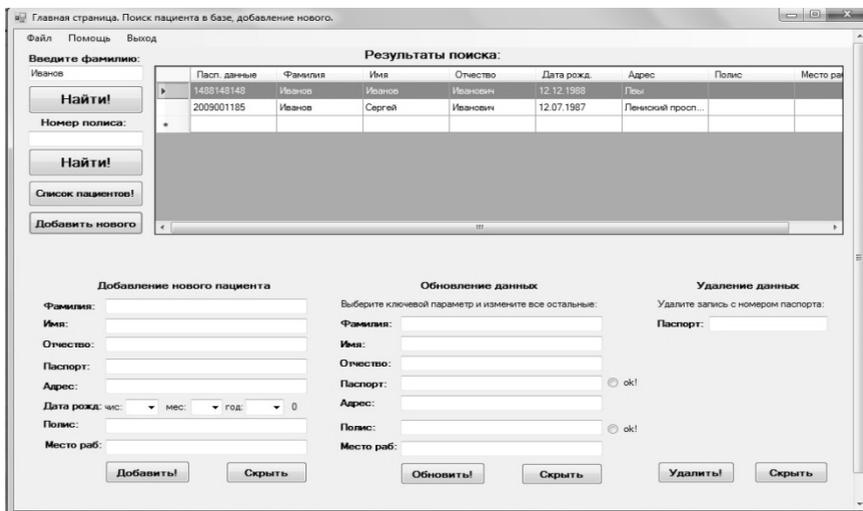


Рис. 2. Главная форма приложения

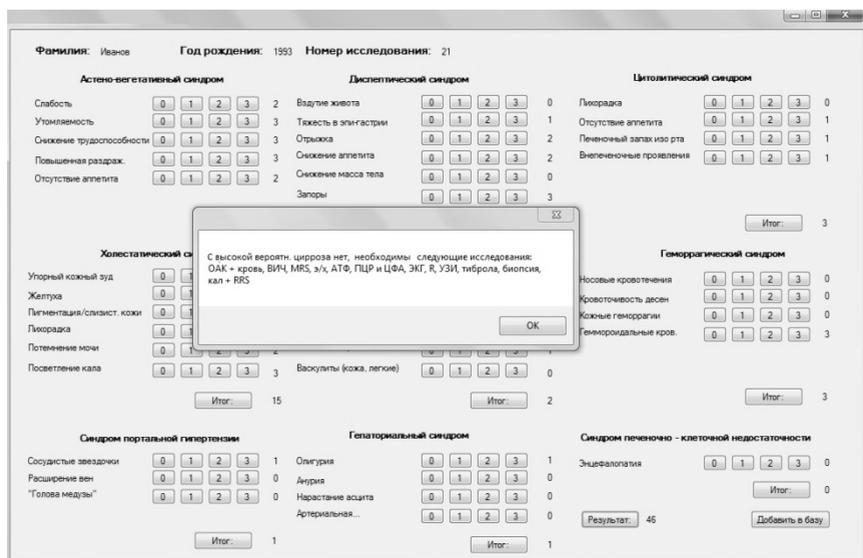


Рис. 3. Модуль программы с обучающей частью

Заключение

В рамках данной работы был создан программный комплекс для диагностики инфекционных заболеваний с включенной в него единой

базой данных пациентов с инфекционными заболеваниями. В ходе реализации были решены следующие задачи:

- создана база данных, позволяющая вносить и редактировать сведения о пациентах, хранить результаты анализов, сохранять и редактировать историю болезни пациента;

- реализована простейшая обучающая система для помощи врачу в постановке правильного диагноза.

Список литературы

1. Гусев А.В. Медицинские информационные системы [Электронный ресурс] : обзор, изменения, тренды. – Режим доступа : http://www.kmis.ru/site.nsf/pages/medsoft2012_1.htm

2. Учайкин В.Ф. Инфекционные болезни у детей [Электронный ресурс] / В.Ф. Учайкин, В.И. Нисевич, О.В. Шамшева. – Режим доступа : <http://www.studmedlib.ru/book/ISBN9785970423417.html>

3. Гильманов А.А. Медицинские информационные системы [Электронный ресурс] : использование информационных медицинских систем в управлении лечебно-профилактическим учреждением. – Режим доступа : <http://cyberleninka.ru/article/n/opyt-prepodavaniya-meditsinskoj-informatiki-s-ispolzovaniem-meditsinskih-informatsionnyh-sistem-primenyaemyh-v-meditsinskih>

4. Справочное руководство по MySQL [Электронный ресурс] : интернет-справочник. – Режим доступа : <http://www.mysql.ru/docs/man/Reference.html>

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА ВРАЧА- ЭПИДЕМИОЛОГА

А.Б. Пилюгин

Студент

В.В. Колесник

Интерн

С.В. Борзунов

Доцент

С.Д. Кургалин

Профессор

Н.В. Габбасова

Профессор

Введение

Природно-очаговые инфекции продолжают оставаться одной из актуальных проблем патологии населения России. Освоение прежде малозаселенных территорий поставило перед медицинскими службами ряд неотложных практических задач, связанных с существованием природных очагов инфекционных заболеваний. Для их решения необходимо составить представление о наличии на осваиваемых территориях природно-очаговых заболеваний, оценить опасность этих болезней для населения, определить формы, темп и направления изменений природных очагов под влиянием хозяйственной деятельности человека. Актуальность изучения природно-очаговых инфекций в Воронежской области обусловлена не только отсутствием достоверной регистрации случаев инфекционных заболеваний среди населения и несовершенством их диагностики, неполным охватом декретированных групп населения вакцинопрофилактикой (т.е. профилактикой инфекционных болезней путем вакцинации), но и несовершенством мониторинга и прогноза этих заболеваний.

Для решения сформулированных выше задач в настоящей работе был создан программный комплекс, позволяющий автоматизировать рабочее место врача-эпидемиолога и с его помощью оценить влияние погодных условий на популяцию переносчиков заболеваний. Данный комплекс даёт

© Пилюгин А.Б., Колесник В.В. (ВГМУ им. Бурденко),
Борзунов С.В., Кургалин С.Д., Габбасова Н.В. (ВГМУ им.
Бурденко), 2016

возможность одновременно учитывать несколько параметров, делать прогнозы относительной численности не только популяций мелких млекопитающих и клещей, но и отдельных видов переносчиков болезней в зависимости от погодных условий. Информация об аналогичных свободно распространяемых программах, используемых в Воронежской области, в настоящее время отсутствует.

1. Цель и задачи исследования

– Собрать, упорядочить и проанализировать данные о численности популяции животных и насекомых – переносчиков болезней и о погодных условиях в Воронежской области за последние годы.

– Определить, существует ли корреляция между численностью переносчиков болезней и погодными условиями.

– Выполнить краткосрочный (на полгода) прогноз численности животных – переносчиков болезней по Воронежской области.

В ходе проведённого исследования были использованы находящиеся в открытом доступе сведения о погоде в Воронежской области за последние 16 лет [1] : о температуре воздуха, об уровне осадков и величине снежного покрова. Информация о численности животных – переносчиков болезней была нам предоставлена ФБУЗ «Центр гигиены и эпидемиологии» (данные с 2006 г. по 2015 г.). Эта информация помещена в базу данных (БД), из которой она в дальнейшем извлекается для последующей обработки.

Характеристики погодных условий и численность животных отображаются в зависимости от времени. На рис. 1 кривая «домовая мышь» представляет зависимость относительной численности мышей соответствующего вида от времени. Кривой «домовая мышь – прогноз» является функция регрессии для популяции данного вида. Таким образом, в краткосрочном прогнозе можно увидеть динамику численности вида. Пунктирными кривыми обозначены границы доверительной области для функции регрессии.

Важное значение для прогнозирования имеет величина корреляции, для расчёта которой использовался метод Пирсона [2]. Кроме того, вычислялась и доверительная вероятность, позволяющая судить о наличии корреляции.

На рис. 2 представлен интерфейс модуля расчёта коэффициента корреляции. Видно, что коэффициент корреляции между температурой окружающей среды и размером популяции желтогорлых мышей равен 0.84, и упомянутые величины находятся в прямой корреляционной зависимости с достоверностью 0.99.

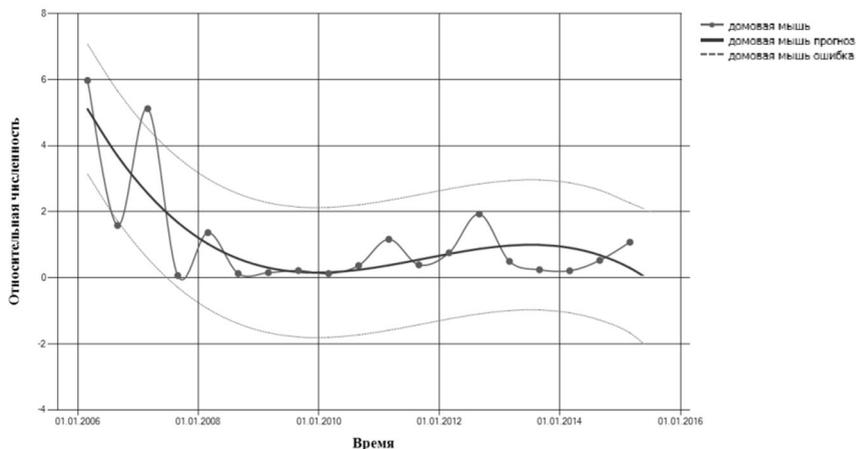


Рис. 1. График временной зависимости численности вида «домовая мышь» и её регрессионная кривая

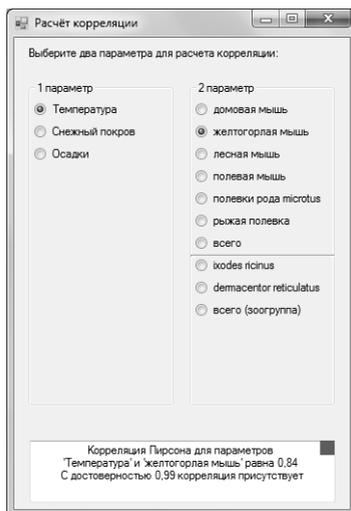


Рис. 2. Один из результатов расчёта корреляции между погодными характеристиками и популяцией вида

2. Реализация программного комплекса

Приложение было реализовано на языке C#. Используется база данных под управлением СУБД SQLite ver.3.

Данные о численности животных – переносчиков болезней обрабатываются автоматически из предоставляемых ФБУЗ «Центр гигиены и эпидемиологии» файлов (в формате .xlsx) с помощью библиотеки Excel [3]. Обработка проходит в три этапа. Первый этап – чтение из файла, второй – запись полученных данных в программные сущности, третий – сохранение данных в БД.

Рассмотрим схему базы данных (см. рис. 3). В ней составным ключом (fromDate, toDate) обозначены определённые периоды, в которые производится отлов млекопитающих (сущность population).

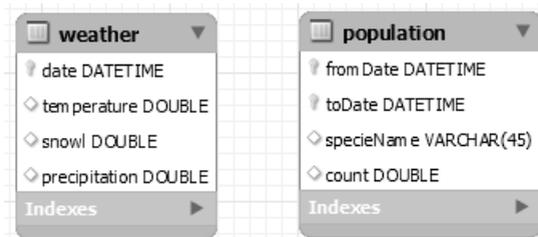


Рис. 3. Схема базы данных, используемой в приложении

Рассмотрим метод расчёта корреляции параметров погодных условий и численности животных в программном комплексе. Аргументы функции для расчёта этой величины представлены в программе в виде списков (популяции млекопитающих/насекомых и погодных условий). Список, содержащий погодные условия, формируется следующим образом: берутся даты измерения численности животных (fromDate, toDate) и находится среднее значение погодных показателей, поступивших на вход функции в качестве второго аргумента, в промежуток времени между fromDate и toDate.

Коэффициент корреляции Пирсона ρ рассчитывается по формуле [2]:

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (1)$$

где x_i , y_i – случайные величины ($i = 1, 2, \dots, n$).

Для оценки достоверности полученного результата необходимо определить параметр

$$r_{\alpha} = \sqrt{\frac{t_{1+\alpha}^2}{n-2+t_{1+\alpha}^2}}, \quad (2)$$

где t_{α} – альфа-квантиль распределения Стьюдента с $n-2$ степенями свободы.

В свою очередь, для нахождения альфа-квантилей используется функция обратного распределения Стьюдента *invstudenttdistribution* библиотеки *alglib* [4].

Для получения прогноза численности животных используется функция *FinancialFormula* класса *DataManipulator* [5]. Её синтаксис: *FinancialFormula(FinancialFormula.Forecasting, "RegressionType,Period,ApproxError,ForecastError", "Historical", "UpperError,LowerError");*

Используются следующие параметры:

RegressionType

тип регрессии (линейная, экспоненциальная, логарифмическая, степенная);

Period

период прогнозирования;

ApproxError

логический параметр, показывающий, выводится ли погрешность приближения при построении регрессии. Если установлено значение **false**, то выходной ряд погрешностей не содержит данных. Значение этого параметра по умолчанию – **true**;

ForecastError

логический параметр, который показывает, выводится ли погрешность прогнозирования численности животных. Если для этого параметра установлено значение **false**, то выходной ряд погрешностей не содержит ошибку приближения для всех прогнозируемых точек данных при условии, что параметр *ApproxError* имеет значение **true**. Значение этого параметра по умолчанию – **true**;

UpperError

погрешность верхней границы прогнозирования;

LowerError

погрешность нижней границы прогнозирования.

В зависимости от типа регрессии функция прогнозирования будет выглядеть по-разному. Так, при линейной регрессии, эта функция будет иметь вид $ax+b$. Экспериментальным путём было выяснено, что логарифмическая регрессия даёт наименьшую ошибку прогнозирования, поэтому в программе используется именно она.

3. Результаты работы программы

Результаты, полученные с использованием программного комплекса, показали, что для одних видов животных существует прямая корреляция с погодными условиями (+), для других – обратная (-), а для некоторых корреляция отсутствует (см. таблицу).

Таблица

Наличие корреляции между погодными условиями и видами животных

	Температура	Осадки	Снег
Домовая мышь			
Желтогорлая мышь	+		+
Лесная мышь	+		+
Полевая мышь	+		+
Полевки рода <i>Microtus</i>	+		
Рыжая полевка	+		+
Всего мышей	+		
Клещи вида <i>Ixodes ricinus</i>			-
Клещи рода <i>Dermacentor</i>	-		
Всего клещей	-		

Как следует из таблицы, повышение температуры положительным образом сказывается на численности, например, желтогорлой мыши, в то время как при значительном снежном покрове их популяция будет меньше средней. Кроме того, было выявлено, что количество осадков не влияет ни на численность клещей, ни на численность мышей.

Расчеты по прогнозированию показали, что после резкого увеличения численности клещей в ближайшие месяцы (в мае-июне 2016 г.) ожидается резкий её спад, летом 2016 г. также ожидается небольшое уменьшение численности мышей.

Заключение

В настоящей работе создан программный комплекс, который обеспечивает:

1. Хранение в базе данных и эффективное использование информации о погодных условиях в Воронежской области с 2000 г. по настоящее время и численности животных и насекомых – переносчиков болезней с 2005 г. по 2015 г.

2. Определение наличия корреляции между популяцией животных и насекомых – переносчиков болезней и погодными условиями.

3. Построение краткосрочного прогноза численности переносчиков инфекционных болезней.

Разработанный программный комплекс является первоначальным вариантом автоматизированного места врача-эпидемиолога. Усовершенствованная и дополненная версия данного программного комплекса позволит повысить эффективность его деятельности, так как предоставит ему возможности по более точному прогнозированию на больший период времени.

Список литературы

1. Погода в мире [Электронный ресурс] : Архив погоды в Воронеже. – Режим доступа: [http://rp5.ru/Архив_погоды_в_Воронеже_Чертовицком_\(аэропорт\),_METAR](http://rp5.ru/Архив_погоды_в_Воронеже_Чертовицком_(аэропорт),_METAR)
2. Кобзарь А.И. Прикладная математическая статистика / А.И. Кобзарь. – М. : Физматлит, 2006. – 813 с.
3. Github [Электронный ресурс] : ExcelDataReader. – Режим доступа : <https://github.com/ExcelDataReader/ExcelDataReader>
4. ALGLIB [Электронный ресурс] : О проекте ALGLIB. – Режим доступа : <http://alglib.sources.ru/#about>
5. MSDN – сеть разработчиков Microsoft [Электронный ресурс] : Формула прогнозирования. – Режим доступа : [https://msdn.microsoft.com/ru-ru/library/dd456655\(v=vs.100\).aspx](https://msdn.microsoft.com/ru-ru/library/dd456655(v=vs.100).aspx)

СОДЕРЖАНИЕ

1. А.В. Аносова, Д.И. Соломатин. ПРИЛОЖЕНИЕ ДЛЯ СОЗДАНИЯ СТЕРЕОИЗОБРАЖЕНИЙ.....	3
2. В.С. Бондаренко, А.П. Бубнов, П.С. Лысачев. АНАЛИЗ РЕЗУЛЬТАТОВ ОПРОСОВ С ПОМОЩЬЮ ТАБЛИЦ СОПРЯЖЕННОСТИ, ФАКТОРИЗАЦИИ НЕОТРИЦАТЕЛЬНЫХ МАТРИЦ И ЛОГАРИФМИЧЕСКИ-ЛИНЕЙНЫХ МОДЕЛЕЙ.....	8
3. И.А. Долгобородов, Д.И. Соломатин. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ЗАДАЧ ПО ЯЗЫКУ SQL.....	15
4. Д.А. Душкин, С.А. Зуев. АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ МИКРОКЛИМАТОМ В ЗАКРЫТЫХ ПОМЕЩЕНИЯХ.....	21
5. М.В. Землянухин, Д.И. Соломатин. ФРЕЙМВОРК ДЛЯ ПОДГОТОВКИ СОРЕВНОВАНИЙ ПО ПРОГРАММИРОВАНИЮ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА.....	28
6. И.В. Капранчиков, Я.А. Туровский, С.Д. Кургалин. ИНФОРМАЦИОННАЯ СИСТЕМА ЖИЗНЕОБЕСПЕЧЕНИЯ БИОЛОГИЧЕСКОЙ ТКАНИ.....	35
7. Д.А. Капустина, А.В. Атанов. АЛГОРИТМ НОРМАЛИЗАЦИИ УРАВНЕНИЙ МНОГОМЕРНЫХ ПРОСТРАНСТВ.....	40
8. В.В. Карнович, Н.А. Тюкачев. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ МАГАЗИНОВ РОЗНИЧНОЙ СЕТИ ООО «ФОРМАТ».....	45
9. А.С. Колесов, Д.И. Соломатин. ПОИСК ПОХОЖИХ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ НА ОСНОВЕ АНАЛИЗА АУДИОСИГНАЛА.....	51
10. М.М. Корчагин, С.А. Вялых. АНАЛИЗ РИСКОВ ИСПОЛЬЗОВАНИЯ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ.....	57
11. В.П. Кузнецова, А.С. Миронова. ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА ЛЕКАРСТВЕННЫХ ПРЕПАРАТОВ.....	64
12. А.Г. Ласкин, С.А. Вялых. МОНИТОРИНГ ЦЕЛОСТНОСТИ ВЕБ-САЙТА.....	70
13. П.О. Левчук, С.А. Зуев. УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ПО РАСПИСАНИЮ.....	75
14. Р.В. Литвинов, Д.Н. Борисов. РЕАЛИЗАЦИЯ СЕНСОРНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ МИКРОКОНТРОЛЛЕРОВ MSP430.....	81
15. М.С. Мишанин, Д.И. Соломатин. ПОСТРОЕНИЕ РОБОТА НА БАЗЕ МИКРОКОНТРОЛЛЕРА RASPBERRY PI.....	90

16. П.А. Морозов, С.А. Зуев. ИЗУЧЕНИЕ РЕЧЕВЫХ ТЕХНОЛОГИЙ И ПОСЛЕДУЮЩЕЕ ИХ ПРИМЕНЕНИЕ В СФЕРЕ ЗАПИСИ РЕЧЕВОЙ ИНФОРМАЦИИ.....	97
17. Е.Ю. Морозов, А.В. Лобода. СИСТЕМЫ БИЛИНЕЙНЫХ УРАВНЕНИЙ	106
18. В.А. Назарьева, Т.В. Новицкая. ИНТЕГРИРОВАНИЕ МАТРИЧНЫХ АЛГЕБР ЛИ.....	112
19. А.А. Ногих, Д.И. Соломатин. ТЕКСТОНЕЗАВИСИМАЯ ИДЕНТИФИКАЦИЯ ДИКТОРА ПО ГОЛОСУ.....	117
20. Е.Ю. Обущенко, М.А. Дрюченко. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ КОНТРОЛЯ АУТЕНТИЧНОСТИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ.....	124
21. А.Б. Ольферук, Н.А. Тюкачев. РАЗРАБОТКА МОДЕЛЕЙ И АЛГОРИТМОВ ПРОСТРАНСТВЕННОЙ НАВИГАЦИИ ВНУТРИ ПОМЕЩЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ ИВЕАСОН.....	130
22. А.Г. Плотникова, И.В. Абрамов РЕАЛИЗАЦИЯ МЕТОДИКИ ПРОФИЛАКТИКИ ПРОЦЕССНЫХ РИСКОВ.....	136
23. Е.В. Погребецкая, А.В. Атанов. АФФИННО-ОДНОРОДНЫЕ ГИПЕРПОВЕРХНОСТИ ПАРАБОЛИЧЕСКОГО ТИПА С ИНДЕФИНИТНОЙ ФОРМОЙ ЛЕВИ.....	141
24. М.В. Полев, А.В. Сычев. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ ПОЛУЧЕНИЯ ДАННЫХ ИЗ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»	147
25. А.С. Пономарев, П.С. Лысачев. РАЗРАБОТКА МОДУЛЕЙ АНАЛИЗА ДАННЫХ И ОТМЕНЫ ДЕЙСТВИЙ ДЛЯ ПРОГРАММНОГО КОМПЛЕКСА DYNAMICS.....	154
26. С.В. Саввин, А.А. Сирота. МОДЕЛИ И АЛГОРИТМЫ СУПЕРПИКСЕЛЬНОЙ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ.....	160
27. А.А. Смолин, А.В. Сычев. ИНТЕРАКТИВНЫЙ ИНСТРУМЕНТ ДЛЯ ВИЗУАЛЬНОГО АНАЛИЗА МНОГОМЕРНЫХ ДАННЫХ НА ОСНОВЕ ПАРАЛЛЕЛЬНЫХ КООРДИНАТ.....	166
28. С.В. Смотров, В.В. Гаршина. ОЦЕНКА УРОВНЯ ИНФОРМАЦИОННОЙ ЗАЩИЩЕННОСТИ ОРГАНИЗАЦИИ НА ОСНОВЕ БАЙЕСОВСКИХ СЕТЕЙ ДОВЕРИЯ	173
29. Д.М. Суходолов, А.А. Крыловецкий. ПОСТРОЕНИЕ СВЕРХРАЗРЕШЕНИЯ ПО ОДНОМУ ИЗОБРАЖЕНИЮ.....	179
30. С.И. Тюленев, Н.Н. Винокурова. ПРОЕКТИРОВАНИЕ СЕТЕЙ ПОДВИЖНОЙ СВЯЗИ С КОДОВЫМ РАЗДЕЛЕНИЕМ КАНАЛОВ	186

31. А.Г. Чурсин, Н.А. Тюкачев, Д.И. Соломатин. ПРИМЕНЕНИЕ ТЕХНОЛОГИИ НАDOOP ДЛЯ АНАЛИЗА "БОЛЬШИХ ДАННЫХ" НА ПРИМЕРЕ ЗАДАЧИ ГЕОАНАЛИТИКИ.....	191
32. А.В. Швырева, А.А. Сирота. РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ СОЗДАНИЯ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ НА ОСНОВЕ ДЕФОРМИРУЮЩИХ ПРЕОБРАЗОВАНИЙ ИЗОБРАЖЕНИЯ.....	198
33. А.В. Щеглов, В.Ю. Храмов. СПОСОБ РАЗРАБОТКИ ТЕСТОВЫХ УДАЛЕННЫХ ИНФОРМАЦИОННО-ТЕХНИЧЕСКИХ ВОЗДЕЙСТВИЙ НА ПРОСТРАНСТВЕННО-РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ИНФОРМАЦИОННО-ТЕХНИЧЕСКИХ СРЕДСТВ.....	203
34. O. Alkaadi, M. Khoder, S.V. Vlasov. USING DATA MINING METHODS TO OPTIMIZE OVERBOOKING	211
35. M. Khoder, O. Alkaadi, S.V. Vlasov IMPROVED HEURISTIC LOAD BALANCING ALGORITHM.....	216
36. Г.В. Измайлов, Т.Ю. Брусенская, П.В. Саенко, С.В. Борзунов, С.Д. Кургалин, Ю.Г. Притулина. ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ДИАГНОСТИКИ ИНФЕКЦИОННЫХ ЗАБОЛЕВАНИЙ.....	224
37. А.Б. Пилюгин, В.В. Колесник, С.В. Борзунов, С.Д. Кургалин, Н.В. Габбасова. ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА ВРАЧА-ЭПИДЕМИОЛОГА.....	230

Научное издание

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

Под редакцией Д. Н. Борисова

Выпуск 10

Подп. в печ. 15.06.2016. Формат 60x84/16. Усл. печ. л. 14,0. Тираж 500 экз. Заказ 495.

Издательский дом ВГУ
394000, г. Воронеж, пл. им. Ленина, 10
Отпечатано с готового оригинал-макета в типографии
Издательского дома ВГУ.
394000, г. Воронеж, ул. Пушкинская, 3