

УДК 004.6

А.А. Козинский, В.А. Козлов

ПРИМЕНЕНИЕ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ ДЛЯ АНАЛИЗА ЗВУКОВЫХ ДАННЫХ

Анализ данных различной природы получил широкое распространение в разных областях практической деятельности. Вычислительная техника позволяет решать задачи анализа данных в криминалистике, криптографии, системе обучения, распознавания образов и других. В статье описана система для анализа звуковых данных. В основу математической модели положено разложение звуковой волны на частотные характеристики с использованием алгоритма быстрого преобразования Фурье. Описанная математическая модель реализована в приложении, используемом для обучения иностранных студентов русскому языку.

На современном этапе развития информационных технологий проблема анализа и сравнения обрабатываемых данных является одной из самых важных в сфере обработки информации. Так, например, распределенная сеть дата-центров крупнейшей в мире компании Google общей мощностью 220 МВт уже в 2006 году проводила анализ порядка 70 ТБ-изображений для сервиса Google Earth и 800 ТБ-данных поисковой базы веб-документов с уровнем компрессии 11% [1, 2].

Распознавание текстовой информации на растровых изображениях, идентификация пользователя в целях защиты от несанкционированного доступа к приватным системам на основе анализа спектральных характеристик голоса, анализ и определение уникальности цифровых отпечатков пальцев, радужной оболочки глаза или голоса физического лица – вот лишь небольшой список задач, требующих активного использования знаний из теории анализа, методов численного решения задач на электронно-вычислительных машинах и математической статистики.

В настоящей работе особое внимание уделяется математическим методам обработки звуковых данных и их реализации средствами программирования. В частности, рассмотрена задача сравнения двух наборов звуковых данных с целью поиска качественных и количественных параметров для определения степени их подобия.

Постановка задачи: реализовать программный проект для определения степени подобия двух отрезков звуковой дорожки, включающий математическую модель и ее программную реализацию; использовать результат проектирования для усовершенствования существующих технологий поиска и защиты информации, управления данными; дополнительно предусмотреть в программной реализации модели возможности сетевых технологий управления данными с целью применения результатов для систем дистанционного обучения. Таким образом, задача сравнения звуковых данных является прикладной задачей, которая может быть компонентом других систем.

В нашем случае основной системой, использующей анализ звуковых данных, был выбран тренажер для обучения русскому языку иностранных студентов. Большинство проанализированных нами программных систем подобного рода обладают возможностью предоставления учебной информации (содержания обучения) в текстовой или мультимедийной (звуковой) форме. Обучающийся имеет возможность прочесть интересующий его материал, прослушать лекционную запись. Современные веб-технологии предоставляют массу средств для создания и использования тренажеров с описанными возможностями: многочисленные CMS (например, ModX), программные фреймворки (jQuery, Mootools), системы тестирования (ИнфоТест). Однако возможно-

сти организации диалога между системой и студентом чаще всего ограничены только представлением материала.

Проверочные задачи, тесты, опросы и реализация проверки результатов их выполнения достаточно широко освещены в литературе. Основная идея традиционного сравнения – это реализация анализа текстовой информации, введенной пользователем. Такие решения реализованы в ряде систем, например Moodle.

В нашем случае сравнению подлежат звуковые данные. Для решения задачи используются эталонные, заведомо верные наборы аудиозаписей, которые необходимо сравнить с полученным набором данных (назовем его «входным»). При реализации математической модели и ее реализации возникают трудности, связанные с формой представления информации: использование значительных вычислительных мощностей для обработки большого объема данных, необходимость выполнять вычисления в режиме реального времени, нетривиальность используемого алгоритма сравнения данных и другие.

Опишем основные идеи, использованные для моделирования сравнения звуковых данных, которые могут быть представлены несколькими этапами.

Первый этап. При распознавании образца проводится первоначальное трансформирование вводимой информации для сокращения обрабатываемого объема и представления информации в виде, удобном для анализа.

Следующим (вторым) этапом является спектральное представление аудиоинформации образца. Спектральное представление достигается путем использования широкополосного анализа записи.

Хотя спектральное представление звука позволяет повысить эффективность его обработки, необходимо помнить, что исходный сигнал (до момента спектрального представления) весьма разнообразен. Разнообразие возникает по многим причинам, а именно:

- индивидуальные различия человеческих голосов;
- особенности различных уровней речи одного и того же лица в разные моменты времени;
- вариации в произношении говорящего лица фраз и их частей;
- особенности варьирования движений естественных артикуляторов говорящего (языка, губ, челюсти, нёба).

С учетом изложенных особенностей на втором этапе определяются конечные выходные параметры для варьирования голоса и производится нормализация. На основе нормализованной речи составляются шкалы параметров, определяется ситуационный уровень речи говорящего.

Определенные параметры речи используются в последующем (на третьем этапе) для создания шаблона. Шаблон включается в словарь, который характеризует произнесение звуков при передаче информации говорящим, являющимся источником входного сигнала.

На четвертом этапе после распознавания новые входные речевые образцы (также нормализованные и параметризованные) сравниваются с шаблонами, уже имеющимися в базе данных. При сравнении используется динамичное искажение и другие метрические измерения.

Подробнее остановимся на одном из этапов сравнения данных. Основным этапом сравнения звуковых записей является спектральное преобразование полученных сигналов. Ключевую роль в этом процессе играет преобразование Фурье.

Преобразование Фурье – операция, сопоставляющая одну совокупность функций вещественной переменной с другой функцией вещественной переменной. Эта новая функция описывает коэффициенты («амплитуды») при разложении исходной

функции на элементарные составляющие – гармонические колебания с разными частотами. Поскольку любой звук раскладывается на синусоидальные волны, то вполне объяснимо использовать преобразование Фурье для построения частотного спектра звука. Спектр частот звуковой волны представляет собой график зависимости амплитуды от частоты.

Таким образом, применив для двух звуковых волн преобразование Фурье для получения спектров частот двух волн и предварительно отфильтровав их от лишних шумов (например, используя «окно Кайзера»), на выходе процесса получаем два набора основных гармоник звуковых данных. Непосредственное сравнение уровней звука записей по времени влечет за собой огромные погрешности, которые не позволяют достоверно сравнить две звуковые волны. По указанной причине для анализа полученных показателей нами использованы спектры частот.

Непосредственное использование преобразования Фурье в случае программного анализа невозможно, поскольку вычислительные машины имеют дело только с дискретными наборами данных. Поэтому в работе алгоритма сходства звуковых волн применяется дискретный вариант преобразования Фурье. В этом случае используются предварительно разбитые на несколько фрагментов звуковые записи одинаковой длины со значениями уровня звука соответствующих волн.

Формула дискретного преобразования Фурье имеет вид:

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-ikn2\pi}{N}},$$

где $k = \overline{0..N-1}$, x_n – значения амплитуд начальной волны, а X_k – значения амплитуд спектра. Таким образом, зависимость «амплитуда – время» переходит в зависимость «амплитуда – частота» полученного спектра.

Алгоритм, реализующий код дискретного преобразования Фурье имеет сложность $O(N^2)$. Указанная сложность соответствует малоэффективной реализации и требует существенной доработки. Такая доработка реализована по аналогии с быстрым преобразованием Фурье. Мы под этим преобразованием понимаем один из быстрых алгоритмов, называемый алгоритмом прореживания по частоте/времени или алгоритмом по основанию 2 [3].

Данный алгоритм имеет более приемлемую сложность $O(N \log N)$. Впервые он был опубликован в 1965 году в статье Кули (Cooley) и Тьюки (Tukey) [4]. В основе алгоритма лежит идея, что если одно преобразование на N точек заменить вычислением двух преобразований на $N/2$ точек, то это приведет к уменьшению количества операций в два раза. Таким образом, если взять количество точек дискретизации равным 2^K , мы можем повторить подобную процедуру рекурсивно K раз, что во столько же раз увеличит производительность процесса вычисления. Когда достигается момент времени, при котором невозможно деление отрезка на две равные части, полученные значения объединяются в обратном порядке. Для объединения значений берется комплексно-сопряженный спектр, к нему выполняется быстрое преобразование Фурье, и результат повторно подвергается комплексному сопряжению. Таким образом,

$$\tilde{X}_k = \sum_{n=0}^{N-1} x_n e^{\frac{-ikn2\pi}{N}}, \text{ где } k = \overline{0..N-1}.$$

После реализации алгоритмов быстрого преобразования Фурье для входной звуковой волны и эталонной получаются два набора данных («массива») $X[0..N-1]$

и $Y[0..N-1]$, которые содержат значения спектральной мощности волн. В качестве меры сходства двух звуковых фрагментов используется следующая формула:

$$f_{xy} = \frac{\sum_i (x_i - M_x)(y_i - M_y)}{\sqrt{\sum_i (x_i - M_x)^2} \sqrt{\sum_i (y_i - M_y)^2}},$$

где M_x и M_y – математические ожидания для соответствующих массивов. Полученное значение является коэффициентом корреляции и лежит в промежутке от -1 до 1 включительно. Именно этот коэффициент является конечным показателем сходства двух звуковых данных. Визуально представление сравниваемых звуковых волн показано на рисунке 1. В верхней части рисунка отображена эталонная волна, в нижней – входная, содержащая шумовые помехи. Графики зависимости амплитуды звуковой волны от времени позволяют визуализировать многие приложения, например Adobe Audition.

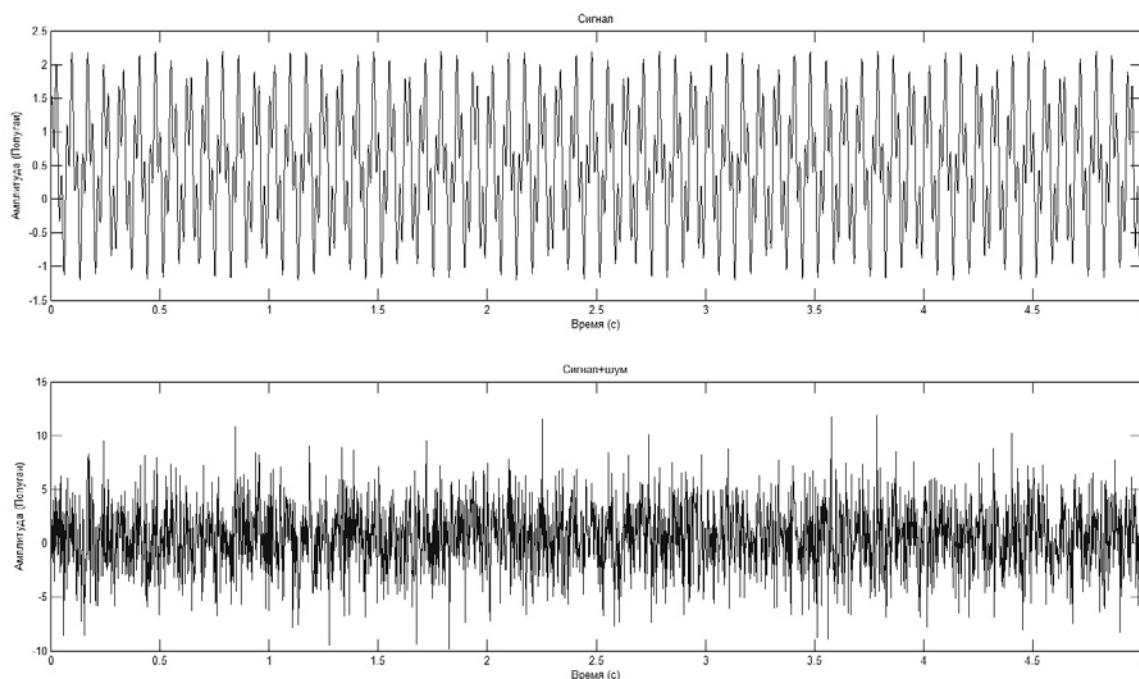


Рисунок 1 – Амплитуды эталонной и входной волны

Способ вычисления меры сходства двух фрагментов, представленных в виде спектра, является наиболее оптимальным. Для решения поставленной задачи анализа сходства звуковых волн нами использованы нейронные сети. Для корректировки шаблона на основе нейронных сетей применена самоорганизующаяся карта признаков Кохонена [5].

На рисунке 2 представлены спектры сравниваемых волн. Сравниваемые волны, эталонная и входная (см. рисунок 1), обрабатываются приложением, реализованным автором. Приложение предназначено для преобразования звуковой волны из представления «амплитуда – время» (рисунок 1) в представление «амплитуда – частота» (рисунок 2). Ниже приведен фрагмент его реализации. Основной метод реализует использование быстрого преобразования Фурье, описанное выше. Результатом работы метода является массив преобразованных данных, содержащий значения спектральных частот волн. Приложение реализовано на языке C++.

Несмотря на различие двух графиков, легко заметить два ярко выраженных пика частот, по которым и определяется степень схожести двух записей (значения остальных частот являются шумом при записи).

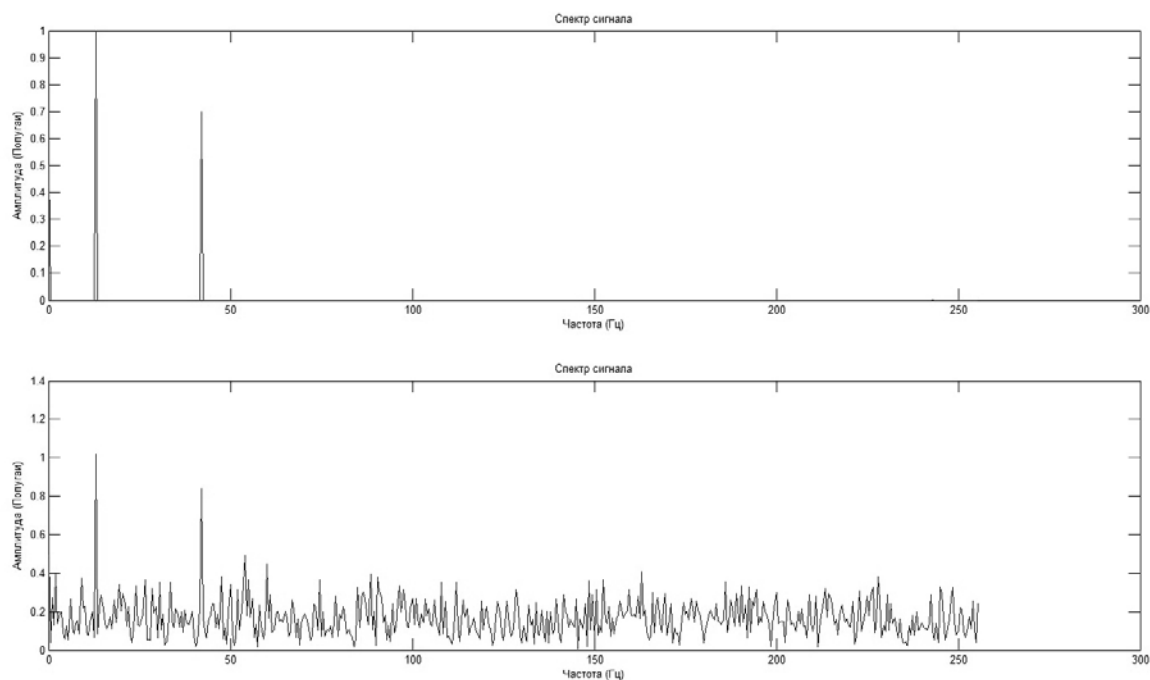


Рисунок 2 – Спектры эталонной и входной волны

Фрагмент программного кода реализации алгоритма быстрого преобразования Фурье на языке программирования C++:

```
#define NUMBER_IS_2_POW_K(x)  (((!(x)&((x)-1)))&&((x)>1))
#define FT_DIRECT             -1    // Direct transform.
#define FT_INVERSE            1     // Inverse transform.

bool FFT(float *Rdat, float *Idat, int N, int LogN, int
Ft_Flag)
{
    // parameters error check:
    if((Rdat == NULL) || (Idat == NULL))                return
false;
    if((N > 16384) || (N < 1))                          return
false;
    if(!NUMBER_IS_2_POW_K(N))                          return
false;
    if((LogN < 2) || (LogN > 14))                      return
false;
    if((Ft_Flag != FT_DIRECT) && (Ft_Flag != FT_INVERSE)) return
false;

    register int  i, j, n, k, io, ie, in, nn;
    float         ru, iu, rtp, itp, rtq, itq, rw, iw, sr;
```

```

static const float Rcoef[14] =
{
    -1.0000000000000000F,  0.0000000000000000F,
0.7071067811865475F,
    0.9238795325112867F,  0.9807852804032304F,
0.9951847266721969F,
    0.9987954562051724F,  0.9996988186962042F,
0.9999247018391445F,
    0.9999811752826011F,  0.9999952938095761F,
0.9999988234517018F,
    0.9999997058628822F,  0.9999999264657178F
};

static const float Icoef[14] =
{
    0.0000000000000000F, -1.0000000000000000F, -
0.7071067811865474F,
    -0.3826834323650897F, -0.1950903220161282F, -
0.0980171403295606F,
    -0.0490676743274180F, -0.0245412285229122F, -
0.0122715382857199F,
    -0.0061358846491544F, -0.0030679567629659F, -
0.0015339801862847F,
    -0.0007669903187427F, -0.0003834951875714F
};

nn = N >> 1;
ie = N;
for(n=1; n<=LogN; n++)
{
    rw = Rcoef[LogN - n];
    iw = Icoef[LogN - n];
    if(Ft_Flag == FT_INVERSE) iw = -iw;
    in = ie >> 1;
    ru = 1.0F;
    iu = 0.0F;
    for(j=0; j<in; j++)
    {
        for(i=j; i<N; i+=ie)
        {
            io      = i + in;
            rtp      = Rdat[i]  + Rdat[io];
            itp      = Idat[i]  + Idat[io];
            rtq      = Rdat[i]  - Rdat[io];
            itq      = Idat[i]  - Idat[io];
            Rdat[io] = rtq * ru - itq * iu;
            Idat[io] = itq * ru + rtq * iu;
            Rdat[i]  = rtp;
            Idat[i]  = itp;
        }
    }

    sr = ru;

```

```

    ru = ru * rw - iu * iw;
    iu = iu * rw + sr * iw;
}

ie >>= 1;
}

for(j=i=1; i<N; i++)
{
    if(i < j)
    {
        io      = i - 1;
        in      = j - 1;
        rtp     = Rdat[in];
        itp     = Idat[in];
        Rdat[in] = Rdat[io];
        Idat[in] = Idat[io];
        Rdat[io] = rtp;
        Idat[io] = itp;
    }

    k = nn;

    while(k < j)
    {
        j = j - k;
        k >>= 1;
    }

    j = j + k;
}

if(Ft_Flag == FT_DIRECT) return true;

rw = 1.0F / N;

for(i=0; i<N; i++)
{
    Rdat[i] *= rw;
    Idat[i] *= rw;
}

return true;
}

```

Решение поставленной задачи имеет важный социальный аспект: в учебном процессе для иностранных студентов, изучающих русский (белорусский) язык, появляется мощное средство фонетического сравнения, позволяющее повысить эффективность изучения нового языка. Как следствие автоматизации – экономия времени, возможность внедрения новых форм обучения, например дистанционного.

Кроме обучения, полученный результат анализа звуковых данных может быть применен в биометрических системах безопасности. В таких системах основным способом персонификации пользователя является указание его сетевого имени и пароля. Опасности, связанные с использованием пароля, хорошо известны: пароли забывают, хранят в неподходящем месте, наконец, их могут просто украсть. Некоторые пользователи записывают пароль на бумаге и держат эти записи рядом со своими рабочими станциями. Как сообщают группы информационных технологий многих компаний, большая часть звонков в службу поддержки связана с забытыми или утратившими силу паролями [6]. Альтернативой текстовой форме ввода имени и пароля может служить звуковая авторизация. В данном случае в качестве пароля выступает фраза владельца системы, которая сравнивается с заранее записанной на момент регистрации фразой. Это позволит избежать ряда серьезных опасностей и неудобных для пользователя моментов. Подобный метод получения доступа к данным не избавляет от распространенных ошибок, а, возможно, даже создает новые. В этом случае эффективным решением может быть использование комбинированных методов защиты информации.

СПИСОК ЛИТЕРАТУРЫ

1. Коваленко, К. Распределительная сеть дата-центров мирового масштаба – сердце империи Google / К. Коваленко [Электронный ресурс]. – Режим доступа : <http://habrahabr.ru/company/mediagrus/blog/167879/>. – Дата доступа : 13.04.2013.
2. Chitu, A. How much data does Google store / A. Chitu [Электронный ресурс]. – Режим доступа : <http://googlesystem.blogspot.com/2006/09/how-much-data-does-google-store.html>. – Дата доступа : 02.09.2006.
3. Быстрое преобразование Фурье // Википедия [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Быстрое_преобразование_Фурье.
4. Программирование. Быстрое преобразование Фурье [Электронный ресурс]. – Режим доступа : <http://psi-logic.shadanakar.org/fft/fft.htm>.
5. Самоорганизующаяся карта Кохонена // Википедия [Электронный ресурс]. – Режим доступа : http://ru.wikipedia.org/wiki/Самоорганизующаяся_карта_Кохонена.
6. Один звонок каждые две секунды и 50 ПБ управляемых данных: такова статистика сервисов управления Fujitsu за 2012 год [Электронный ресурс]. – Режим доступа : http://www.storagenews.ru/news_take.asp?Code=734. – Дата доступа : 29.08.2012.
7. Joseph, P. Методы моделирования сигнала в распознавании речи / P. Joseph, P. Ruslan. – Кемерово, 2000. – 79 с.

A.A. Kozinskiy, V.A. Kozlov. Usage of Fast Fourier Transformation for Sound Data Analysis

Analysis of data of different nature is widespread in a variety of practice areas. Computer science allows us to solve the problem of data analysis in forensics, cryptography, the education system, pattern recognition and others. The article describes a system for analyzing the audio data. The basis of the mathematical model is based on the decomposition of the sound wave at the frequency characteristics with the use of fast Fourier transform algorithm. The mathematical model is implemented in the application used for the training of foreign students Russian language.

Рукапіс паступиў у рэдкалегію 25.04.2013