МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Тверской государственный университет»

Факультет прикладной математики и кибернетики

Кафедра информационных технологий

**ОТЧЕТ**

**по итогам научно-исследовательской работы**

**во 2-ом семестре**

Направление: 02.04.02 «Фундаментальная информатика и

информационные технологии»

Магистерская программа: «Информационные технологии в

управлении и принятии решений»

Тема: «Обработка звука в реальном времени

на платформе Android»

|  |  |  |
| --- | --- | --- |
|  | **Выполнил:**  студент 1 курса магистратуры  Иванов Артём Николаевич  **Научный руководитель:**  кандидат физ.-мат. наук,  Сорокин Сергей Владимирович  **Оценка:** | |
|  |  |

Тверь – 2017 г.

**ОГЛАВЛЕНИЕ**

[**Введение** 2](#_Toc483774546)

[**Цели и задачи** 2](#_Toc483774547)

[**Актуальность** 3](#_Toc483774548)

[**1.** **Проделанная работа** 4](#_Toc483774549)

[**1.1.** **Задача измерения расстояния** 4](#_Toc483774550)

[**1.1.1.** **Нахождение смещения между двумя звуковыми фрагментами** 4](#_Toc483774551)

[**1.1.2.** **Аппаратная задержка** 5](#_Toc483774552)

[**1.1.3.** **Программа-прототип для нахождения расстояния** 7](#_Toc483774553)

[**1.2.** **Задача измерения скорости ветра** 9](#_Toc483774554)

[**Заключение** 11](#_Toc483774555)

# **Введение**

В современном мире наметилась тенденция к компактности: уменьшение размеров устройств, переход к минимализму в дизайне. Большинство социальных сетей и бизнес-коммуникаций переходят в мобильные приложения, для измерения пульса и других биометрических показателей можно пользоваться небольшими часами или браслетами – wearables. Аналогичным образом развиваются технологии и инструментарии для исследовательских экспериментов – к примеру, измерений показателей давления, влажности окружающей среды и т.д. Гораздо проще и удобнее использовать единое устройство для выполнения всех этих замеров, при этом с небольшими затратами и минимальным использованием стороннего оборудования. Достижению этой цели способствуют устройства под операционной системой Android, являющиеся на данный момент самыми распространенными и доступными на мобильном рынке. Смартфоны, использующие различные сенсоры, становятся портативными установками для исследования и измерения различных показателей окружающей среды, одни из таких характеристик могут быть получены посредством обработки звука.

## **Цели и задачи**

В данной научно-исследовательской работе поставлена цель: разработать приложение на платформе Android для обработки звука, которое выполняет следующие функции:

* измерение расстояния;
* измерение скорости ветра.

Для достижения данной цели необходимо выполнить ряд задач:

* ознакомиться с необходимыми материалами и статьями по обработке аудио сигналов;
* разработать предварительно приложение на Python для создания первоначального алгоритма обработки звука и вычислений;
* изучить библиотеки для обработки звука на Android.

## **Актуальность**

Исследования по данной теме могут быть использованы при создании подвижных мобильных игр. Данный тип развлекательных приложений включает физические действия игрока в процессе игры.

Конечное приложение послужит хорошим инструментом при исследовании пещер. Установив его в смартфон можно будет измерить скорость ветра и расстояние, не пользуясь дополнительным оборудованием.

На основании получаемых результатов можно создать дешевый компактный аналог профессиональной установки для измерения соответствующих замеров и характеристик окружающей среды.

# **Проделанная работа**

## **Задача измерения расстояния**

За время научно-исследовательской работы в 1 семестре были изучены статьи на английском языке из исследовательского центра Microsoft по проблеме измерения расстояния с помощью звука. Во втором семестре были изучены мобильные аналоги таких приложений. На текущий момент есть один платный аналог на IOS: Acoustic ruler. С помощью проигрывания звука, приложение может вычислить расстояние между двумя смартфонами IPhone или же между IPhone и подсоединенным к нему наушником. В описании и роликах к программе показали, что возможно измерить расстояние вплоть до 1 метра. Таким образом была обоснована возможность решения данной задачи.

После этого было принято решение создать прототип программы для ПК. Для этого использовался язык программирования Python.

## **Нахождение смещения между двумя звуковыми фрагментами**

В основе первой и второй задачи лежит проблема определения смещения между двумя аудио-фрагментами. В прошлом семестре данное смещение находилось вручную с помощью сопоставления графиков, отображение которых получено с помощью инструментов IDE для Python - Spider. В этом семестре данный процесс был автоматизирован.

Исходные данные: есть две аудиозаписи, которые содержат одинаковые (или похожие) аудио фрагменты, так что: позиции, с которых начинаются эти фрагменты – разные. Время записей (кол-во отсчетов) одинаковое.

Требуется: найти разницу данных позиций.

Решение: с помощью корреляции двух последовательностей (исходных аудиозаписей) находим пик корреляции. Разница позиций (смещение записей одной относительно другой) будет равняться: разнице между позициями: пика и середины корреляции.

Для наглядности представляем рисунок 1:

Сверху вниз:

* Исходные графики со смещением;
* После нахождения смещения: сопоставление графиков;
* Корреляция и ее пик.

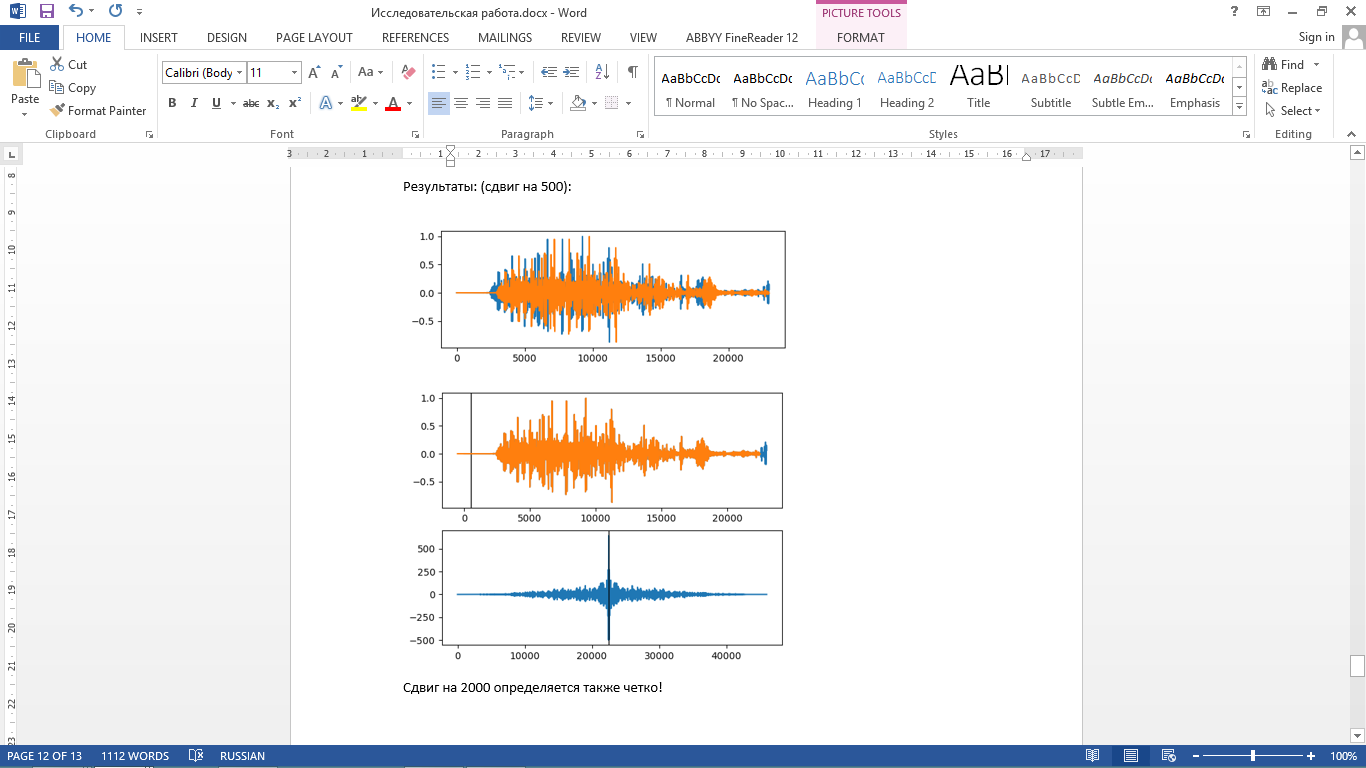


Рисунок 1 Автоматизированный процесс сопоставления графиков

## **Аппаратная задержка**

Задача нахождения расстояния заключается в измерении времени, за которое звук доходит от динамика до микрофона. Так как в нормальных комнатных условиях (температура≈20˚C, воздух, нормальное давление)

скорость звука ≈ 343 м/с, то зная время, мы найдем искомое расстояние.

Допустим, подключаем к ноутбуку динамик, с достаточной длиной провода, чтобы динамик можно было отодвинуть на расстояние до 2-х метров. Тогда, если с помощью программы одновременно проиграть аудиофайл через динамик и записать микрофоном (между микрофоном и динамиком будет некоторое фиксированное расстояние), то как раз получим искомое время. Это будет время между началом и позицией, где начинается сам аудио фрагмент в записи. Чтобы найти эту позицию, будем использовать исходный аудиофайл и решение, описанное выше для нахождения смещения.

Ознакомившись с модулем SoundDevice для Python, мной была написана программа, которая одновременно запускает процесс записи и воспроизведения аудиофайла. В нее было добавлен алгоритм нахождения смещения между двумя звуковыми фрагментами. Таким образом, получили: recAndPlay.py.

Программа берет аудио фрагмент: engSpeech: “Synchronized”, проигрывает через динамик и одновременно записывает через микрофон. Для проверки демонстрируются графики исходной и записанной аудиодорожки на первой паре графиков. А также демонстрируются графики корреляции и совмещения двух записей при полученной задержке.

При первом опыте, оказалось, что задержка слишком велика.

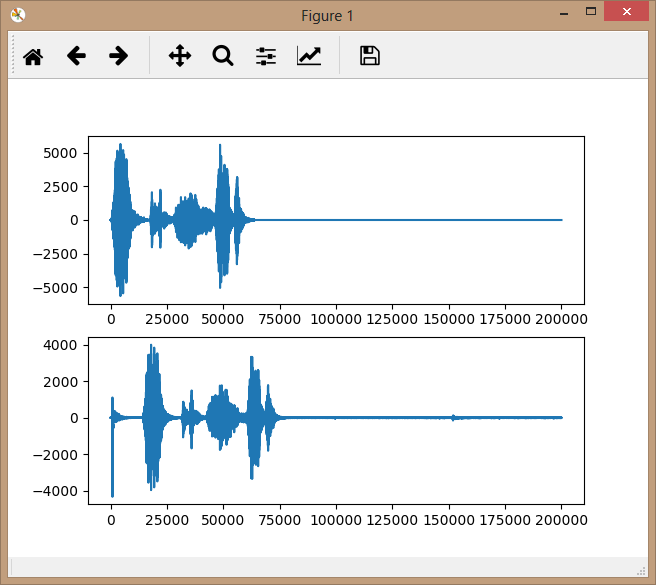


Рисунок 2 Верхний график - оригинал, нижний – запись

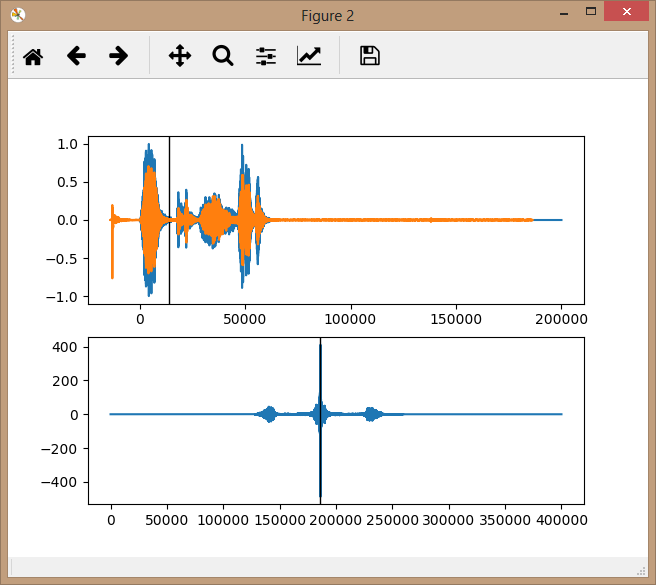


Рисунок 3 На верхнем графике - совмещение записей, на нижнем - их корреляция

Задержка (смещение графиков звуковых фрагментов относительно друг друга) в отсчетах получилась: 13776.5

Далее простыми действиями получим время в секундах: 13776.5/48000=0.2868

И расстояние (зная, что обычная скорость распространения звука = 343 м/с):

0.2868\*343 = 98,37 м.

Что неверно, так как наше расстояние = 0,63 м.

Далее были проведены еще три опыта, при которых динамик стоял на расстоянии 1 см от микрофона. Получились следующие задержки:

1) 13744.5

2) 13676.5

3) 13752.5

Самый большая разница между записями: |13676.5 – 13752.5| = 76 отсчетов.

Проведя необходимые вычисления, получаем:

(76/48000)\*343 = 0.543 м.

Таким образом, столкнулись с проблемой аппаратной задержки, которая не постоянна и имеет большую погрешность.

## **Программа-прототип для нахождения расстояния**

Чтобы избавиться от проблемы аппаратной задержки, воспользуемся следующим ходом действий:

Есть запись аудио файла, который проигрывает один фрагмент1, затем продолжительная пауза и фрагмент2.

Сначала делают запись калибровочную: динамик во время нее находится все время рядом с микрофоном.

Далее делают запись для измерения: здесь при фрагменте1 динамик находится около микрофона, а потом его переносят на расстояние, которое необходимо измерить.

Данные манипуляции также проверялись вручную в 1-ом семестре, во 2-ом для их автоматизации была написана программа с графическим интерфейсом: finalProg.py (Рис.4).

Главное окно программы визуально поделено на две части: «Подготовка» и «Эксперимент».

Сначала пользователь работает с первой частью:

1. выбирает аудиофайл формата wav с соответствующим содержанием (фрагмент1, продолжительная пауза, фрагмент2). Пользователь также может выбрать сохраненную запись аудиофайла в формате txt;
2. Выполняется калибровка: динамик ставится на расстояние 1см от микрофона ноутбука и выполняется одновременная запись и проигрывание аудио файла. Если пользователь выбрал текстовый файл, то запись не производится, а данные берутся из текстового файла;
3. Индикатор калибровки загорается зеленым – программа готова ко второй части.

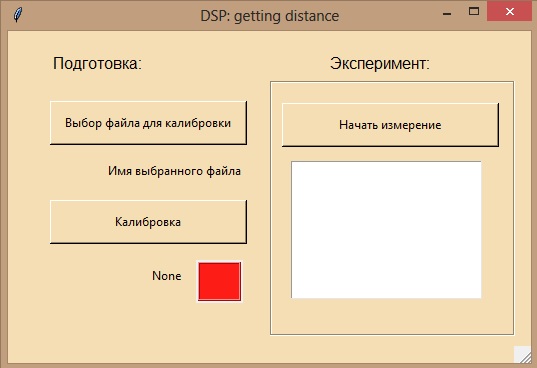


Рисунок 4 Программа для измерения расстояния

«Эксперимент»:

1. Здесь проигрывается та же запись, но после проигрывания фрагмента1, пользователь переносит динамик на некоторое расстояние от микрофона, которое ему нужно измерить. И фрагмент2 уже будет проигрываться не вблизи микрофона.

Далее, с помощью калибровочной записи, синхронизируем аудиозаписи по фрагменту1: вычисляем задержку между фрагментом1 калибровочной записи и фрагментом1 экспериментальной записи.

Совместив графики по фрагменту1, аналогично ищем задержку по фрагменту2.

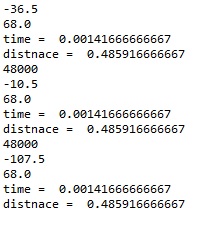
Эта задержка является искомой и означает время, за которое звук проходит искомое расстояние от динамика до микрофона.

Результат измерения и расстояние записывается в элемент отображения «список» под кнопкой «Начать измерение».

Для эксперимента и проверки работы программы провел опыт:

Измерение расстояния в 0,29 м (от микрофона) с помощью finalProg.py (Рис.5).

Как видно по скриншоту, на данный момент, погрешность измерения составляет 4 см. Однако, программа показывает и высокую точность, например: результат при расстоянии 0,49 м (написан в лог-журнал):



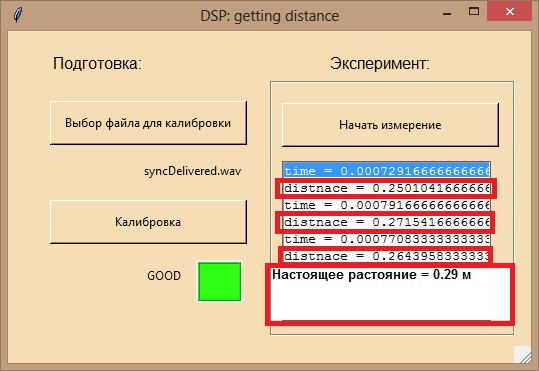


Рисунок 5 Результаты 3-х измерений

## **Задача измерения скорости ветра**

Измерение скорости ветра выполняется следующим способом. Пусть у нас есть микрофон, который находится на ветру и динамик, который проигрывает определенный звуковой фрагмент. Данный динамик мы передвигаем на некоторое расстояние от микрофона навстречу ветру и проигрываем звуковой файл, который записывается микрофоном. Повторяем аналогичные действия, отходя в противоположную сторону от микрофона (вслед ветру) на точно такое же расстояние и записываем звуковой файл. При условии, что у нас есть запись проигрываемого звукового фрагмента, мы можем найти разницу во времени, с которой звук из данных двух точек приходит в микрофон. В первом случае – он придет быстрее, а во втором случае – медленнее. Такая разница обусловлена наличием дующего ветра, способствующего или мешающего достижению звуком микрофона. Путем извлечения данных о времени и имеющейся информации о расстоянии, которое проходит звук, мы можем найти скорость ветра.

С целью наглядного представления данных действий для дальнейшей разработки алгоритма был проведен эксперимент для получения образцов звука.

Взятие образцов звука:

1. Поставили три стула. На них расположены объекты (динамики и диктофон): все на одинаковом расстоянии – 166,5 см.
2. Сделана пробная запись (рис.3). Которая построена следующим образом:

2 секунды (нет звука) + звук синхронизации (нота «соль» (основная октава)) + 4 секунды (нет звука) + перебор основной октавы (до «до»). Наименование файла: **«**проба2.wma**».**

1. Запись звука производится следующим образом:
   1. Включаем запись с помощью предустановленного приложения «Диктофон»
   2. Помещаем динамик на расстоянии 14 см перед микрофоном
   3. Проигрываем запись на динамике

Таким образом, задача измерения скорости ветра отличается от предыдущей тем, что динамик нужно перемещать и для калибровочной записи (см. 1.1.3).

# **Заключение**

В этом семестре была проведена работа по разработке прототипа программы, которая реализует основные функции будущего мобильного приложения. В результате, на языке программирования Python была написана программа finalProg.py, которая вычисляет расстояние между динамиком и микрофоном. Также может быть использована для вычисления скорости ветра. Программа оснащена графическим интерфейсом, который реализован с помощью модуля tkinter и редактора page.

В дальнейшем, согласно плану работ, будет произведено тестирование созданной программы-прототипа и ее улучшение. Вслед за этим будут изучены библиотеки для переноса данного алгоритма на платформу Android.