

Федеральное государственное автономное
образовательное учреждение
высшего профессионального образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
подпись инициалы, фамилия
« ____ » _____ 2019 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Мобильный комплекс для слабовидящих
тема

09.04.01 «Информатика и вычислительная техника»
код и наименование направления

09.04.01.06 «Микропроцессорные системы»
код и наименование магистерской программы

Научный руководитель	_____	<u>доцент, кан.техн.наук</u>	<u>М.С. Медведев</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>А.Ю. Шафеуллах</u>
	подпись, дата		инициалы, фамилия
Рецензент	_____	<u>доцент, кан.техн.наук</u>	<u>Е.В Болдырев</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Нормоконтролер	_____		<u>В.И. Иванов</u>
	подпись, дата		инициалы, фамилия

Красноярск 2019

СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области	7
1.1 Анализ существующих решений	7
1.1.1 Системы с эхолокацией.....	7
1.1.2 Навигационная система парус	12
1.1.3 Система распознавания образов OrCum	13
1.1.4 Voice vision.....	14
1.2 Средства разработки программного обеспечения	15
1.2.1 Arduino IDE	15
1.2.2 HTML/CSS.....	16
1.2.3 JavaScript/TypeScript.....	18
1.2.4 Анализ технологий мобильных приложений	19
1.2.4.1 Кроссплатформенные приложения	20
1.2.4.2 Гибридный подход в разработке приложений	21
1.2.5 Выбор фреймворка для создания мобильного приложения	22
1.2.5.1 React Native	22
1.2.5.2 Ionic 3.....	23
1.2.5.3 Фреймворки для реализации мобильного приложения.....	24
1.3 Выводы по главе	24
2 Метод преобразования изображения в звук	26
2.1 Выделение границ областей изображения.....	26
2.1.1 Линейные методы.....	28
2.1.2 Нелинейный метод	30

2.1.3 Статический метод	32
2.1.4 Алгоритм Канни	33
2.2 Формирование звукового ряда на основе визуальных данных	35
2.2.1 Предварительная обработка изображения.....	35
2.2.2 Выделение границ	36
2.2.3 Преобразование изображения.....	36
2.2.4 Блок-схема алгоритма преобразования.....	38
2.3 Выводы по главе	39
3 Разработка мобильного комплекса	40
3.1 Требование к системе.....	40
3.2 Описание алгоритма.....	41
3.3 Структурная схема	44
3.3.1 Определение дальности	44
3.3.2 Описание сервиса Yandex Vision	46
3.3.3 Описание сервиса Yandex SpeechKit.....	46
3.3.4 Интерфейс приложения	48
Заключение	50
Список использованных источников.....	51

ВВЕДЕНИЕ

По статистике ООН, в процентном соотношении количество инвалидов по зрению по отношению к населению земли составляет: полностью слепые 0,55% (39 млн), инвалиды по зрению 3,5% (246 млн).

Известен ряд программно-аппаратных комплексов для решения задачи ориентации в пространстве слабовидящих. Широкое распространение подобных систем подтверждает актуальность выбранной темы.

Предварительный анализ рынка показал, что имеется три основных типа систем. Условно их можно разделить на системы:

- с рабочей областью;
- эхолокацией;
- фото и видео обработкой.

Преимущество систем с рабочей областью состоит в их безотказной работе и точным определением положения в пространстве. Но у этих систем есть существенный недостаток, пользователю такой системы необходимо находиться в зоне покрытия.

Системы с эхолокацией решают задачу определения расстояния до объекта. Недостатком является низкая точность идентификации.

Системы с фото и видео обработкой решают задачу распознавания образов (жестов, предметов, текста). Недостатком таких систем является отсутствие определения удаленности предметов от слабовидящего.

Таким образом, необходим комплекс, который включает в себя ряд функций:

- определение расстояния до объектов (предметов);
- распознавание образов и текста.
- перевод видео потока в звук;

Решение может быть получено при реализации мобильного приложения.

Целью данной работы является разработка мобильного комплекса ориентации в пространстве для слабовидящих.

Для достижения цели необходимо решить следующие задачи:

- проведение анализа существующих систем;
- разработка метода преобразования визуальных данных в звук.
- проектирование мобильного комплекса;
- разработка программного обеспечения.

1 Анализ предметной области

1.1 Анализ существующих решений

1.1.1 Системы с эхолокацией

Известно, что летучие мыши способны перемещаться в слабо освещенном пространстве, не задевая посторонних предметов, а также охотиться в подобных условиях. Ученые наблюдая за летучими мышами в последствии установили, что эти животные издают ультразвук, который отражается от предметов, и воспринимают его. Научно доказано, что эхолокацию как способ ориентации в пространстве используют различные виды животных, какие как тюлени, землеройки, дельфины [1].

Таким образом, можно сделать вывод, что эхолокация представляется эквивалентом зрения для незрячих и слабовидящих животных или для животных, которые перемещаются при слабом освещении, будь то суша или водоем [1].

Эхолокация (эхо и лат. locatio – «положение») – это способ определения положения объекта по времени задержки возвращений отраженных волн от него. Эхолокация (рисунок 1) может быть акустической или радио – это зависит от типа используемых волн [1].

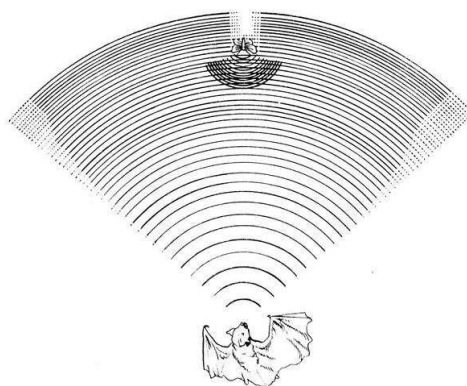


Рисунок 1 – Эхолокация летучей мыши

Сам принцип эхолокации основывается на различной плотности окружающей среды. Сигнал, который издает животное, рассеивается если не встретит на своем пути никаких предметов, но встретив предмет сигнал частично отражается и частично рассеивается. Это зависит от плотности встречаемых объектов.



Рисунок 2 – Принцип действия ультразвукового дальномера

Ультразвуковой дальномер. Ультразвуковой дальномер – устройство для измерения расстояния до предмета, принцип действия которого основан на эхолокации.

Принцип действия наглядно изображен на рисунке 2. В момент измерения создается электрическое колебание при помощи генератора 1, которое преобразуется в ультразвуковую волну 3 при помощи ультразвукового передатчика 2, излучается в окружающее пространство 4. Эта волна отражается от препятствия и возвращается как эхо 5 в приемник 6. Затем в вычислителе 7 происходит обработка полученной информации [1].

Расстояние до препятствия рассчитывается по следующей формуле:

$$d = \frac{v \Delta t}{2}, \quad (1)$$

где d – расстояние до препятствия,

v – скорость звуковой волны,

t – время между отправкой и приемом отраженного сигнала.

Достоинства:

- в отличие от световых дальномеров может определить дальность до стеклянных и бликующих предметов;

- легкость в пользовании;

- приемлемая цена.

Недостатки:

- неоднозначное измерение удаленности от неровной поверхности, это происходит из-за очень большого угла диаграммы направленности;

- точность замера зависит от состояния окружающей среды, которая может меняться с течением времени;

- ограниченное максимальное измерение дальности;

- частичные отражения (паразитный эхосигнал) могут исказить результаты измерений (например, это происходит при высокой влажности);

- измерения объектов из звукопоглощающих, изоляционных материалов или имеющих тканевую поверхность могут привести к неправильным измерениям вследствие поглощения сигнала.

Несмотря на все преимущества и недостатки, ультразвуковой дальномер активно используется слабовидящими в повседневной жизни и варианты его использования могут быть различными. В таблице 1 приведены распространенные системы для слабовидящих построенные на дальномере, имеет смысл проанализировать только одну из приведенных систем, так как они идентичны.

Таблица 1 – Системы с дальномером

Вариант использования	Задача	Оповещение	Крепление дальномера
Тактильная трость с дальномером	Определение дальность до препятствия	Вибрация	Трость
Нагрудный фонарь	Определение дальность до препятствия	Вибрация	Грудь
Электронный поводырь	Определение дальность до препятствия	Вибрация	Рука

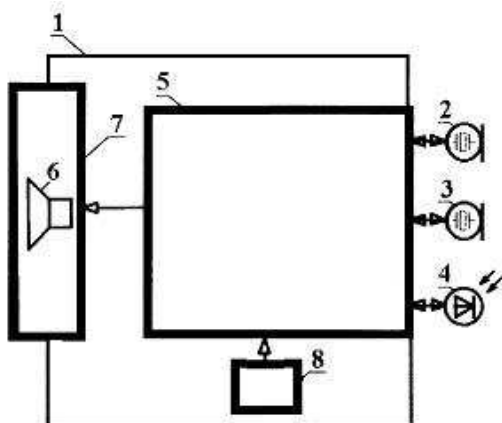
Трость (рисунок 3) для ориентации в пространстве слабовидящих, выполнена в виде тактильной трости и имеет два ультразвуковых дальномеров и один инфракрасный датчик движения, неподвижно установленных на тактильной трости. Взаимное расположение ультразвуковых дальномеров и инфракрасного датчика движения позволяет производить сканирование впереди лежащего пространства ультразвуковым и инфракрасным методом. Первый ультразвуковой дальномер расположен в нижней половине тактильной трости от её свободного конца, второй ультразвуковой дальномер расположен в средней части тактильной трости от её свободного конца. Они установлены так, чтобы их акустические оси были горизонтальными при движении слабовидящего. Инфракрасный датчик установлен в верхней части тактильной трости от её свободного конца. Он установлен так, чтобы его оптическая ось была горизонтально при движении. Информация о препятствиях выводится для слабовидящего в виде голосовых сообщений.



1. Электронный блок
- 2-3. Дальномеры
4. Инфракрасный датчик

Рисунок 3 – Трость

Принцип работы. Как только микропроцессор (рисунок 4) с помощью ультразвуковых дальномеров, обнаружит препятствие на расстоянии 0,8 метра, он подаст аудио сообщение «Препятствие». Если с помощью ультразвуковых дальномеров обнаружит препятствие на расстоянии 0,8 метра, а инфракрасный датчик движения зарегистрирует человека на расстоянии 0,8 метра, микроконтроллер подаст аудио сообщение «Препятствие человек» .



1. Электронный блок обработки информации
- 2-3. Дальномеры
4. Инфракрасный датчик
5. Микропроцессор
6. Динамик
7. Модуль вывода
8. Модуль питания

Рисунок 4 – устройство электронного блока

Недостатком этого технического решения является то, что пользователь должен точно знать о предполагаемом расположении препятствия и заранее сориентировать приемопередатчик на него, либо вести постоянное сканирование путем переориентации приемопередатчика вокруг оси его крепления.

1.1.2 Навигационная система парус

Устройство предназначено для информирования людей с ограниченными возможностями о своем местонахождении, с возможностью получения дополнительных сведений о близлежащих объектах в виде звукового сообщения. Устройство состоит из Базового Блока (рисунок 5) и Активатора (рисунок 6).

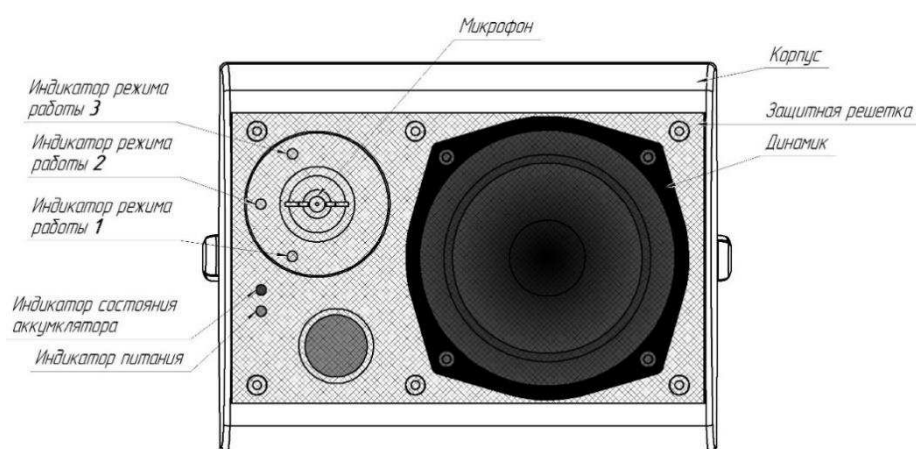


Рисунок 5 – Базовый Блок

Базовый блок предназначен для информирования людей с потерей зрения об их местонахождении с возможностью навигации по территории учреждения (школа, больница, институт, квартира и т.д.) получения дополнительных сведений о близлежащих объектах либо в виде предварительно записанных индивидуальных сообщений. Должна быть предусмотрена возможность работы базового блока совместно с браслетом-активатором на запястье пользователя, который по нажатию на одну из кнопок активирует воспроизведение соответствующего сигнала на базовом блоке системы [2].

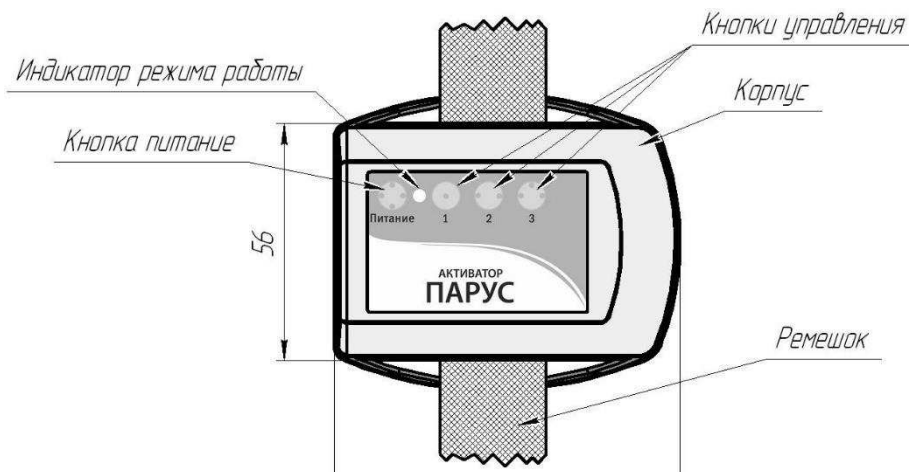


Рисунок 6 – Активатор

Данная система имеет существенный недостаток пользователю такой системы необходимо находиться в зоне покрытия.

1.1.3 Система распознавания образов OrCum

Данная система включает в себя камеру с гарнитурой, которые прикрепляются к очкам, и процессорное устройство по обработке изображений (представлено на рисунке 7).

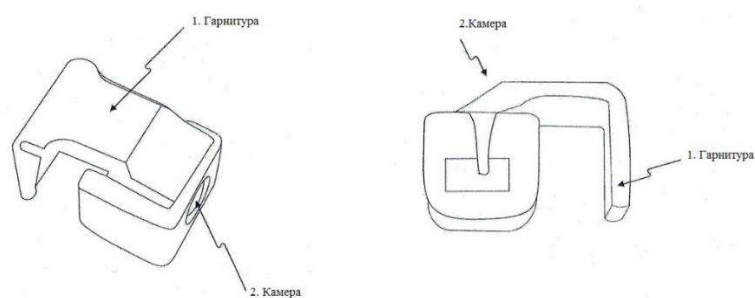


Рисунок 7 – Камера и гарнитура

Принцип работы системы OrCum заключается в следующем, камера фиксирует изображение и передает его устройству обработки, далее устройство определяет на изображении жесты, лица и предметы, которые

были записаны предварительно в память устройства. Затем передает информацию в виде звукового оповещения пользователю.



Рисунок 8 – Вариант использования

Недостатками этой системы является то, что язык поддерживаемый этой системой только английский и нет определения отдалённости предметов от пользователя.

1.1.4 Voice vision

Система Voice vision основана на восприятии закодированной визуальной информации видеокамеры. В сочетании со специализированной программой позволяет слабовидящим при помощи смартфона, оснащенных камерой, слышать звуковой сигнал, сформированный на основе изображения с камеры.

Принцип преобразования изображения (рисунок 9) в звук. Изображения сканируются и озвучиваются по направлению слева направо. По умолчанию скорость озвучивания одного изображения равна одному снимку в секунду. Стереозвук, который слышен, соответственно, прокручивается слева направо,

что облегчает отслеживание направления сканирования изображения. Таким образом, возникший слева или справа от вас звук свидетельствует о наличии зрительного образа, расположенного, соответственно, в левой или правой стороне [2].

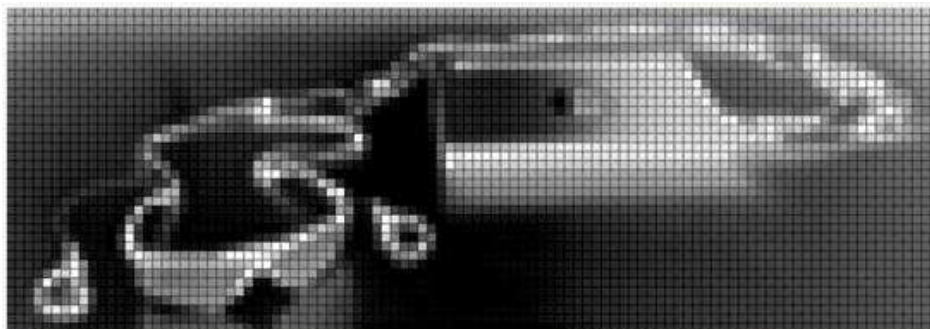


Рисунок 9 – Обработанное изображение

Возникающий при каждом сканировании тон обозначает высоту: чем выше тон, тем выше положение объекта в поле зрения. Следовательно, в зависимости от того, повышается или понижается тон звукового ландшафта, можно судить о том, восходящий, или, соответственно, нисходящий зрительный образ располагается перед вами [2].

Громкость обозначает яркость: чем громче звук, тем ярче (светлее) изображаемый предмет. Следовательно, тишина соответствует черному цвету, громкий звук – белому, а все, что между ними – есть оттенки серого цвета.

Описанный способ позволяет озвучить любое изображение, но чем оно насыщеннее, тем больше звуков потребуется для его отображения. В конечном итоге сигнал, который слышит слабовидящий, звучит как резкий постоянный шум, что является существенным недостатком, так как это может негативно отразиться на центральной нервной системе [2].

1.2 Средства разработки программного обеспечения

1.2.1 Arduino IDE

Arduino IDE – это программное обеспечение для разработчиков платформы Arduino. Эта платформа предназначена для написания программ, их компиляции и программирования аппаратуры [3].

Язык программирования Arduino является стандартным C++ (используется компилятор AVR-GCC) с некоторыми особенностями, облегчающие написание программ новичкам в этом деле.

Возможности Arduino IDE:

- интуитивно понятный и удобный интерфейс;
- программное обеспечение совместимо со всеми версиями операционной системы Windows;
- наличие инструментов необходимых для работы;
- различный набор примеров программ;

1.2.2 HTML/CSS

HTML – специализированный язык разметки, на котором создают веб-документы.

Гипертекст – это некоторый текст, который содержит ссылку на другой текст. В веб сфере эти ссылки являются ссылками на другие веб-документы.

Язык разметки – это набор синтаксических правил, с помощью которых можно упорядочено представить ту или иную информацию на экране. Важно отметить, что язык разметки не является языком программирования [4].

HTML-документ – это веб-документ, который был сформирован при помощи языка разметки HTML. По своей сути – это текстовый файл, в котором был использован язык разметки. Исходный код HTML-документа может быть просмотрен и отредактирован в любом текстовом редакторе, например, в Блокноте. Это одно из удобных свойств HTML-документа. Программа, которая позволяет отображать HTML-документ, на основе исходного кода разметки называется браузером [4].

Браузер интерпретирует HTML-документ, выстраивая его структуру и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения [4].

Процесс интерпретации начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы [4].

CSS (англ. Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего облика документа, который написан при помощи языка разметки [5].

Данный язык используется как средство оформления и описания внешнего облика веб-страниц, написанных с использованием языков разметки HTML и XHTML, а также может использоваться с XML[5]. CSS описывает правила изменения элементов при помощи свойств и ограничения значений этих свойств [5].

Таблица стилей применяется при создании веб-страниц для изменения шрифтов, расположения отдельных элементов, цвета и других свойств элементов страницы. Главной задачей разработки на CSS являлось разделение описания логической структуры веб-страницы (которое производится языком разметки) от описания внешнего вида этой веб-страницы (которое производится формальным языком).

1.2.3 JavaScript/TypeScript

JavaScript – это язык сценариев, при помощи которого создаются диалоговые HTML-документы, также дает возможность производить вычисления и без обращения к серверам проводить процедуру проверки допустимости данных [6].

JavaScript относится к классу объектно-ориентированных языков. В связи с этим необходимо ознакомиться с тремя терминами.

Объект – это предмет такой как: окно браузера, формы и их части, кнопки и текстовые окна. Также JavaScript имеет собственную группу встроенных объектов, к которым относятся массивы, данные.

Метод – это действия, которые может выполнять объект.

Свойства, которые есть у всех объектов.

Сценарий JavaScript в двоичный код не компилируются. Они посредством встроенного в браузер интерпретатора, интерпретируются и обрабатываются.

Как только браузер обнаружит код JavaScript на странице, то он сразу же начинает его выполнять. Все строки кода будут выполняться последовательно, переход к следующей строке происходит только после выполнения предыдущих строк. Если в HTML-документе будет содержаться несколько сценариев JavaScript, то их исполнение будет происходить в порядке расположения в документе[6].

Возможности JavaScript:

- изменять содержимое страницы: редактировать текст, добавлять и удалять теги, редактировать стили элементов;
- события: скрипт может ожидать выполнения определенного действия (клика мышки или нажатие клавиши) и выполнять определенную функцию;
- загрузка данных при обращении к серверу позволяет наполнять страницу без её перезагрузки;

- вывод сообщений, обработка данных пользователя и другие.
- большое количество библиотек для построения различных информационных панелей.

TypeScript, в отличие от JavaScript, имеет возможность поддержки использования полноценных классов (как в классических объектно-ориентированных языках) и явного статического назначения типов. К тому же имеет ещё возможность поддержки подключения модулей, что позволит повысить скорость разработки, облегчить читаемость и повторное использование кода, также может помочь осуществить поиск ошибок на этапе разработки и компиляции, возможно ускорить выполнение написанной программы.

TypeScript, компилируется в JavaScript так как он обратно совместим с ним, поэтому после компиляции программы на TypeScript можно выполнять в любом современном браузере или например использовать совместно с серверной платформой Node.js. Предполагается, что в связи полной обратной совместимости преобразование ранее созданных приложений на новый язык программирования может происходить путём последовательного определения типов.

1.2.4 Анализ технологий мобильных приложений

Native (с англ. Native - "родной") приложения – это программные продукты, которые разработаны под определенную операционную систему. Программа, которая была написана, например, под Android, будет работать только на этой операционной системе.

У каждой операционной системы есть свой набор основных процедур, посредством которых она осуществляет взаимодействие с различными программами прикладного уровня [3].

Разработка Native приложения происходит с определенным API, отсюда следует, что и работать они будут на определенной операционной системе.

Преимущества:

- высокая производительность;
- обработка большого количества данных на стороне клиента;
- гибкая реализация;
- экономичное потребление батареи;

1.2.4.1 Кроссплатформенные приложения

Кроссплатформенное программное обеспечение – это такое программное обеспечение, которое работает на нескольких аппаратных платформах или операционных системах [3].

Кроссплатформенные приложения – создаются на языке разметки HTML и стилей CSS, а также JavaScript. Данные приложения пишутся одновременно для всех платформ и адаптивны к большинству устройств. Большинство разработчиков при разработке кроссплатформенных приложений пользуются фреймворком PhoneGap. Данный фреймворк позволяет получить доступ к аппаратным и программным возможностям платформы.

PhoneGap — бесплатный open-source, созданный Nitobi Software. Позволяет создать приложения для мобильных устройств используя JavaScript, HTML5 и CSS3, без необходимости знания «родных» языков программирования (например, Objective-C, Java), под все мобильные операционные системы такие как Android, iOS и так далее. Готовое приложение компилируется в виде установочных пакетов для каждой мобильной операционной системы [3].

Преимущества кроссплатформенной разработки:

- экономия бюджета проекта, так как используется одна технология и один набор графики, что позволяет снизить количество рабочих часов;

- время разработки, отсутствие уникальных элементов интерфейса позволяет значительно снизить сроки разработки.

Недостатки кроссплатформенной разработки:

- не используются уникальные особенности платформ;
- медленная работа приложения;
- приложению необходим постоянный доступ в интернет, чтобы загружать контент.

1.2.4.2 Гибридный подход в разработке приложений

Гибридное приложение сочетает в себе некоторые функции, что имеет нативное и веб-приложение.

Достоинства гибридного приложения:

- кроссплатформенность. Сделав одно приложение, можно экспортировать его под любую операционную систему – iOS, Android, Windows Phone, BlackBerry;

- доступная стоимость разработки – в разы меньше, чем при нативном подходе;

- доступ к основным данным мобильного устройства: GPS, камера, микрофон и т.д.;

- возможность распространять приложение через официальные магазины приложений.

Недостатки гибридной разработки:

- Сниженная скорость работы в отличие от нативных приложений.
- Проблематичная верстка адаптивного дизайна. Несмотря на наличие различных веб-фреймворков для построения приложений

Ограниченное представление визуальных и графических элементов, в частности, анимации.

1.2.5 Выбор фреймворка для создания мобильного приложения

1.2.5.1 React Native

React Native – это JS-фреймворк, который разработала компания Facebook для разработки мобильных приложений с использованием JavaScript. React Native[14] использует те же самые фундаментальные стандартные блоки UI что и обычные приложения iOS и Android, не используя при этом ни браузер, ни WebView/UI-WebView.

На рисунке 10 представлена архитектура приложения на React Native.

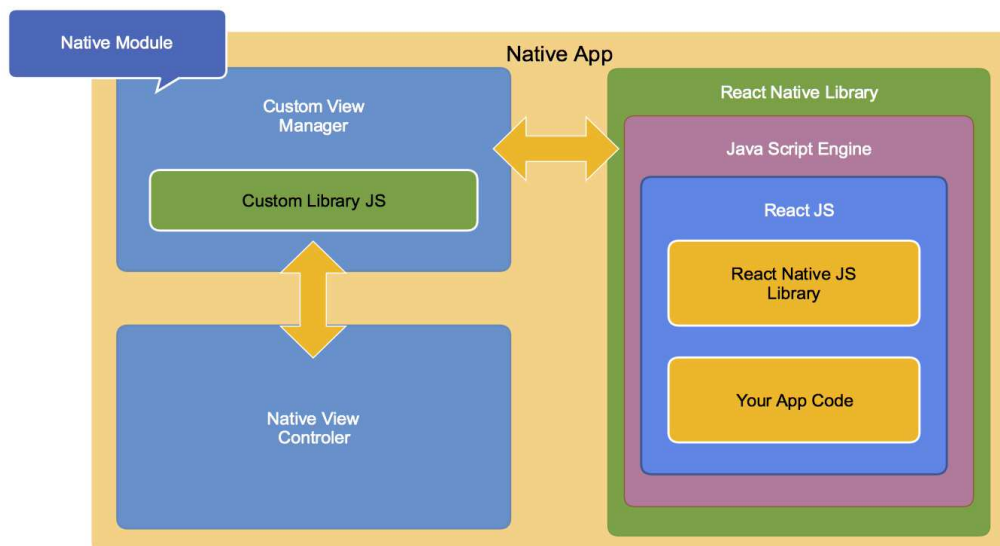


Рисунок 10 – Архитектура приложения на React Native

1.2.5.2 Ionic 3

Ionic Framework – это фреймворк [15] для создания гибридных мобильных приложений. В состав фреймворка входят:

- JavaScript и CSS компонент, созданных на основе AngularJS;
- SASS;
- Apache Cordova.

Ionic Framework – это дополнение над очень популярным фреймворком Apache Cordova [19]. Приложения, разработанные на этом фреймворке, на телефоне выполняются в специальной оболочке, которая позволяет показывать HTML и выполнять JavaScript. Ionic следует шаблону контроллера View-Controller. В шаблоне View Controller существуют разные разделы интерфейса как дочерние контроллеры вида, которые содержат в себе другие виды. далее контроллеры вида воспроизводят области просмотра, находящиеся внутри них, чтобы обеспечить взаимодействие и функциональность пользовательского интерфейса.

На рисунке 11 представлена архитектурная схема приложения Apache Cordova.

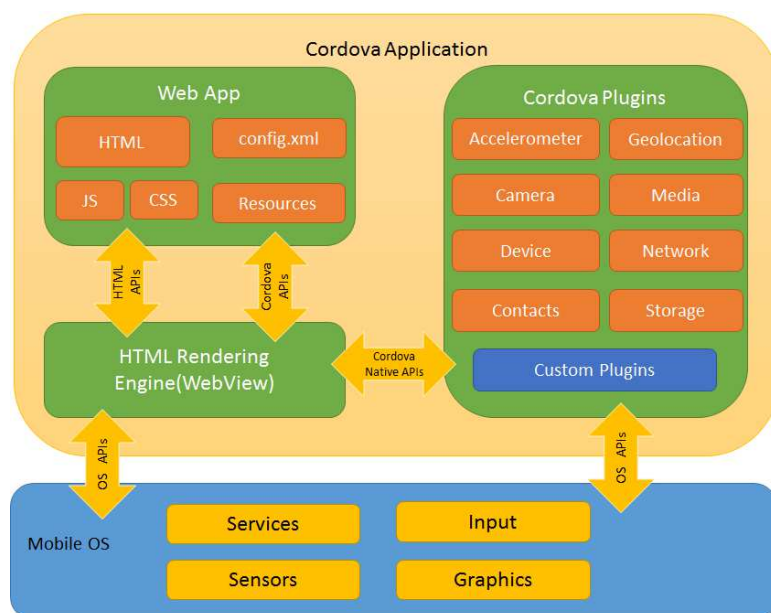


Рисунок 11 – Архитектура приложения на Apache Cordova

Apache Cordova является посредником между пользовательским интерфейсом и ресурсами мобильного устройства. Ionic – это один из фреймворков, который предоставляет UI виджеты. MVC модель для него предоставляет Angular. Native функции мобильного устройства такие как, камера, микрофон, местоположение из браузера недоступны. Поэтому для того что бы работать с ними необходимо использовать плагины Apache Cordova.

1.2.5.3 Фреймворки для реализации мобильного приложения

Выбор фреймворка зависит от поставленных задач, конечной цели и многих других факторов. Использование Ionic 3 позволит быстро разработать прототип мобильного приложения. При использовании React Native можно столкнуться с рядом дополнительных задач. Например, необходимость работы с нативным кодом платформы ObjectiveC/Swift или Java.

Для реализации мобильного комплекса был выбран фреймворк Ionic 3, преимущества данного фреймворка представлены ниже.

- известный набор инструментов для разработчика набор используемых технологий является знакомым, это существенно сокращает время на выполнение поставленной задачи;

- быстрый старт используя шаблоны приложений, предоставленные Ionic, можно за короткое время создать прототип для показа заказчику;

- написанное приложение работает на всех популярных платформах, это дает кроссплатформенность с незначительными изменениями кода.

1.3 Выводы по главе

В результате анализа существующих решений ориентации в пространстве для слабовидящих и методов, на которых они основаны, был

выявлен существенный недостаток – это малый набор функций. Каждая из рассмотренных (таблица 2) систем решает поставленную задачу одним методом или выполняет только одну функцию, что в ряде случаев делает их неэффективными.

Таблица 2 – Сравнительная таблица

Система	Задача	Оповещение	Язык	Стоимость
Трость	Дальность до объекта	Вибрация	—	1,5 тыс.руб
OrCum	Распознавание образов и текста	Аудио сигнал	Английский	4,250 тыс.дол
Парус	Навигация в зоне покрытия (класс, комната и т.п.)	Аудио сигнал	Русский	26,7 тыс.руб
Voice Vision	Замена зрения	Аудио сигнал	—	—

Таким образом, можно сделать вывод, что если система будет основана на нескольких различных методах ориентации в пространстве, то это повысит её эффективность.

2 Метод преобразования изображения в звук

Данная глава посвящена описанию алгоритма и разработанного метода формирования звукового ряда на основе визуальных данных.

2.1 Выделение границ областей изображения

Методы выделения границ изображений основываются на алгоритмах, которые выделяют пиксели цифрового изображения, в которых резко изменяется яркость или есть другие виды неоднородностей.

Большинство алгоритмов, основаны на вычислении градиента. Так как изображение дискретно и в классическом смысле производной в данном случае не существует, то используют вычисление приближенного значения производной [7].

Пространственные фильтры изменяют значения пикселя в зависимости от изменений яркости световой интенсивности его соседних пикселей. Соседние пиксели определяются исходя из размера матрицы, или маски, центр которой приходится на рассматриваемый пиксель.

В случае матрицы 3x3, как показано на нижеследующем рисунке, значение центрального пикселя (показан сплошной заливкой) устанавливается в зависимости от значений восьми окружающих его пикселей (показаны штриховкой) [7].

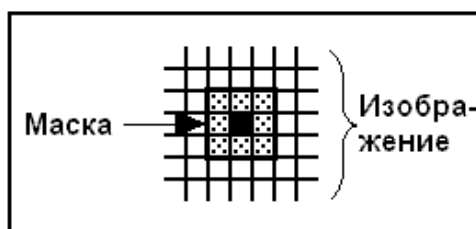


Рисунок 12 – Маска

Некоторые локальные преобразования оперируют одновременно как со значениями пикселей в окрестности, так и с соответствующими им значениями некоторой матрицы, имеющей те же размеры, что и окрестность. Такую матрицу называют фильтром, маской, ядром. Значения элементов матрицы принято называть коэффициентами.

С точки зрения распознавания и анализа объектов на изображении наиболее информативными являются не значения яркостей пикселей областей объектов, а характеристики их границ – контуров. Другими словами, основная информация заключена не в яркости отдельных областей, а в их очертаниях.

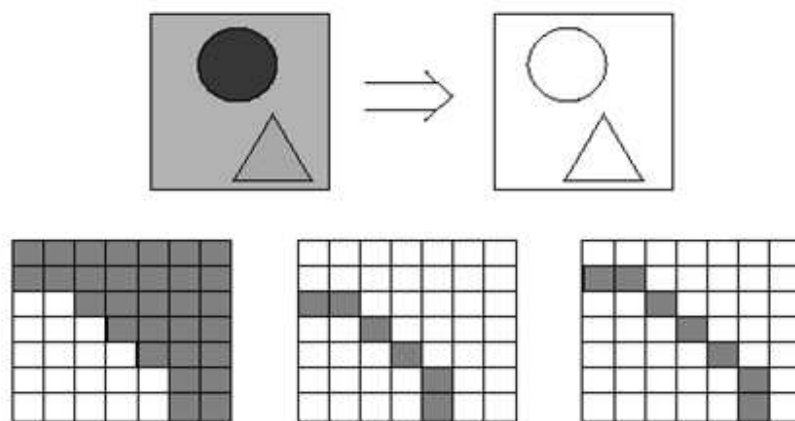


Рисунок 13 – Выделение контура изображения

Контуром изображения называют совокупность его пикселей, в окрестности которых наблюдается скачкообразное изменение функции яркости.

Так как при цифровой обработке изображение представлено как функция целочисленных аргументов, то контуры представляются линиями шириной, как минимум, в один пиксель.

При этом может возникнуть неоднозначность в определении линии контура. Поэтому необходимо использовать дополнительную обработку изображения [7].

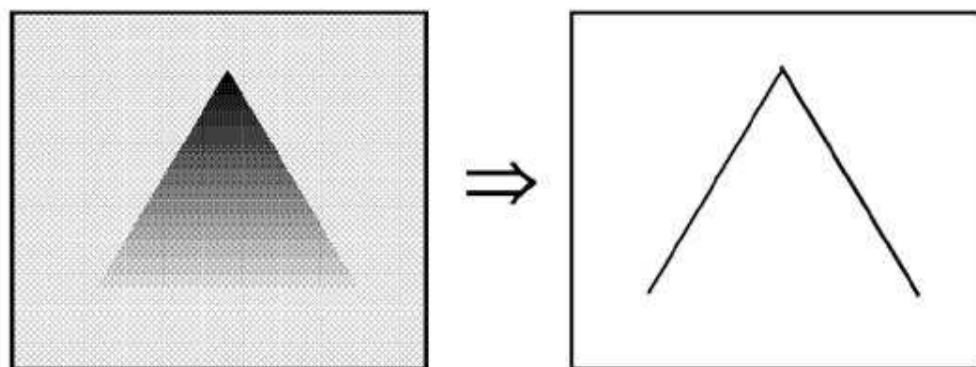


Рисунок 14 – Разрыв контура

Методы выделения границ бывают линейные и нелинейные, рассмотрим основные из них.

2.1.1 Линейные методы

Самым простым линейным алгоритмом, который основан на производных, является оператор Лапласа [11] (для функции двух переменных) $f(x, y)$ определяется как :

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}. \quad (2)$$

Вторая производная, в области цифровой обработки сигналов, по координате x и y рассчитывается по формулам:

$$\frac{\delta^2 f}{\delta x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y), \quad (3)$$

$$\frac{\delta^2 f}{\delta y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y). \quad (4)$$

$$\nabla^2 f = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \quad (5).$$

Это уравнение может быть реализовано с помощью маски представленной на рисунке 12.

0	1	0
1	-4	1
0	1	0

Рисунок 12 – Маска Лапласа

Оператор Лапласа является второй производной и его применение показывает разрывы уровней яркостей, а также может подавлять область со слабым изменением яркостей изображения.

В результате получается изображение, которое содержит серые линии вместо контуров и разрывов, которые находились на затемненном фоне без особых выраженностей. Вообще фон возможно восстановить, сохраняя при этом повышение резкости, достигнутой оператором Лапласа [8].

Надо сложить исходное изображение и оператор Лапласа:

$$f'(x, y) = f(x, y) + L(x, y), \quad (6)$$

где L – преобразование оператором Лапласа.

2.1.2 Нелинейный метод

В цифровой обработке изображений производные первого порядка решаются через модуль градиента. Для функции $f(x, y)$ градиент в точке (x, y) определяется как двумерный вектор-столбец:

$$\nabla^2 f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (7)$$

$$\Delta f = |\Delta f| = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}. \quad (8)$$

На практике часто, чтобы избавиться от нелинейности используют приближенное значение модуля:

$$\nabla f = |G_x| + |G_y|, \quad (9)$$

В маске Собела [11] (рисунок 15) приближенное значение производной по координате x достигается путем разности между третьей и первой строках, а приближенное значение производной по координате y разность между третьем и первом столбцах.

-1	-2	-1
0	0	0
1	2	1

Рисунок 15 – Маска Собела

Также есть маски Робертса (Рисунок 16) и Превитта. Оператор Робертса дает плохие результаты относительно полезного сигнала и шумовой составляющей изображения.

-1	0	0	-1
0	1	1	0

Рисунок 16 – Маска Робертса

Маска Превитта аналогична маске Собела, разница в том, что в маске Превитта стоят 1, у маски Собела 2.

Оператор Уоллеса [12], данный оператор работает с маской, представленной на рисунке 18.

$$\begin{bmatrix} Z_1 & Z_2 & Z_3 \\ Z_4 & F & Z_6 \\ Z_7 & Z_8 & Z_9 \end{bmatrix}$$

Рисунок 18 – Маска Уоллеса

$$F' = \frac{\ln \left[\frac{F}{Z_2} \cdot \frac{F}{Z_4} \cdot \frac{F}{Z_6} \cdot \frac{F}{Z_8} \right]}{4}, \quad (10)$$

Где \ln – натуральный логарифм, новое значение вычисляется по приведенной формуле.

Ещё одним нелинейным методом выделения границ является оператор Кирша:

$$f(x, y) = \max(r_1, \dots, r_8), \quad (10)$$

где r – абсолютные значения перепадов яркости по восьми направлениям окрестности 3×3 пикселя (рисунок 17).

$$\begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

Рисунок 17 – Маска Кирша

Данная маска вращается вокруг центра, следовательно, оператор Кирша позволяет выравнять яркость пикселей для контуров со слабым перепадом яркости [13].

2.1.3 Статический метод

Данный метод является двух этапным и применим для маски произвольной конфигурации [17].

На первом этапе вычисляется среднее значение яркостей пикселей по текущему рабочему окну:

$$\mu = \frac{1}{m \cdot n} \cdot \sum_{i=1}^m \sum_{j=1}^n F(i, j). \quad (11)$$

На втором этапе вычисляется среднеквадратичное отклонение яркостей пикселей в рабочем окне от среднеквадратичного значения:

$$\sigma = \sqrt{\frac{1}{mn} \cdot \sum_{i=1}^m \sum_{j=1}^n (F(i, j) - \mu)^2}. \quad (12)$$

Далее значения всех значение всех элементов рабочего окна умножаются на σ :

$$F' = \sigma \cdot F(i, j). \quad (13)$$

Преимущество статического метода заключается в том, что изменяются значения сразу всех элементов маски, но это является и недостатком так как это увеличивает вычислительную мощность необходимую для выполнения алгоритма

2.1.4 Алгоритм Канни

Оператор Канни [16] не только вычисляет градиент размытого изображения и удаляет точки, которые не являются максимумами. Далее удаляются слабые границы, с использованием двух порогов. Например, если на части изображения нет ни одной точки со значением большим верхнего порога, происходит его удаление.

Также можно включить в алгоритм шумоподавления, что увеличит четкость контурных границ изображения [18].

Общие этапы определения границ методом Канни.

На первом этапе исходное изображение проходит операцию сглаживания с помощью свертки:

$$G_0(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (11)$$

$$g(x, y) = G_0(x, y, \sigma) * f(x, y). \quad (12)$$

где σ – параметр степени сглаживания(чем выше, тем больше сглаживается изображение).

Затем выполняется вычисление градиента в вертикальном и горизонтальном направлениях с помощью операторов первой производной (Робертса, Собела, Прюитта и др.).

Далее полученные значения объединяются с помощью известной формулы:

$$E(i, j) = [(g_V(i, j)^2 + g(i, j)_H^2)]^{1/2}, \quad (13)$$

$$\theta(i, j) = \tan^{-1} [g_x(i, j) / g(i, j)_y], \quad (14)$$

$$E_T(i, j) = \begin{cases} E_{i,j}, & \text{если } E_{i,j} > T \\ 0, & \text{иначе} \end{cases} \quad (15)$$

где T – выбирается таким образом, чтобы все элементы границы были сохранены, в то время как большинство шумов подавлено.

Последним этапом является уточнение найденных на предыдущем шаге контуры путем обнуления значений градиента, которые не находятся на вершине гребня градиента. Операция использует два порога $T1$ и $T2$, причем $T1 > T2$. Операция начинается в точке на гребне, значение которой больше порога $T1$ и продолжается до тех пор, пока высота не станет меньше порога $T2$. Данная процедура предназначена для нахождения значимых контуров изображения.

Таким образом можно сделать вывод, наиболее подходящий метод для поставленной задачи является Канни, базируется на предыдущих и не требует больших вычислительных затрат и к тому же выделяет только значимые контуры, что дает хороший показатель отношение полезного сигнала и шума.

С помощью данного фильтра возможно выделить границы, но использовать его целесообразно только после устранения шумов и перевода изображения в черно-белое пространство.

2.2 Формирование звукового ряда на основе визуальных данных

2.2.1 Предварительная обработка изображения

Предварительная обработка изображения включает преобразование исходного разрешения в формат 1000 на 1000 пикселей. Далее выполняется перевод в цветовой режим (рисунок 17) GrayScale (градации серого).



Рисунок 17 – Предварительная обработка

Это позволяет уменьшить вычислительные затраты, что соответственно отразится на скорости выполнения.

2.2.2 Выделение границ

Одним из самых оптимальных методов нахождения границ, в отношении сигнал скорость обработки, оказался Канни. Фильтр Канни выигрывает у других фильтров, например: фильтра Робертса, фильтра Превитта, фильтра Собеля, метод водораздела, из-за его высокой скорости обработки, удобным выбором коэффициентов для последующей обработки и сохранения найденных контуров, низкой вероятностью ошибки, возможностью анализа пересекающихся контуров.

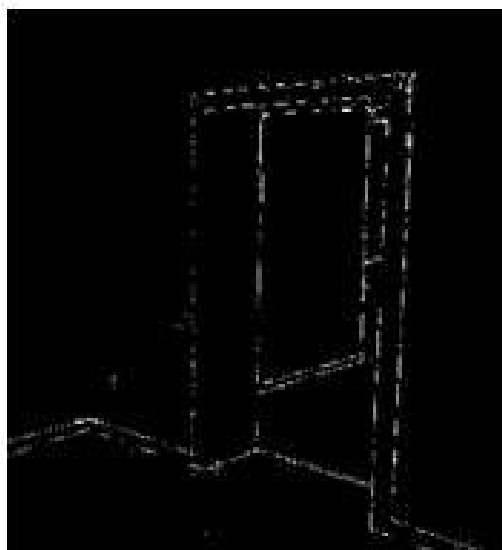


Рисунок 18 – Выделение границ Канни

2.2.3 Преобразование изображения

Для решения преобразования выделенных на предыдущем этапе характеристик изображения, представленных в виде двоичной матрицы, предложен следующий метод.

На первом шаге выполняется построчная разбивка матрицы характеристик на группы векторов, которые будут соответствовать определенным диапазонам частот синтезируемого звукового сигнала (рисунок 19). Основной задачей при выборе определенных значений частот для векторов был поиск интервала между соседними значениями гармоник, которые с одной стороны должны вносить в сигнал информацию о присутствии в визуальном тракте глобальных изменений, а с другой быть достаточно разнесены друг от друга, для устранения наложений.

Для решения данной задачи был использован принцип формирования звукоряда на основе пентатоники. Пентатоникой называют лад, состоящий из 5 нот, находящихся в пределах одной октавы. Преимуществом такого подхода является обеспечение консонанса – гармонического созвучия сочетания различных частот, то есть отсутствия диссонанса [9]. Это важно, так как разрабатываемый программно-аппаратный комплекс рассчитан на продолжительное по времени применение пользователем, и воспроизведение раздражающих слух сигналов может привести к утомлению [10].

В итоге была построена таблица частот, представлена в таблице 3.

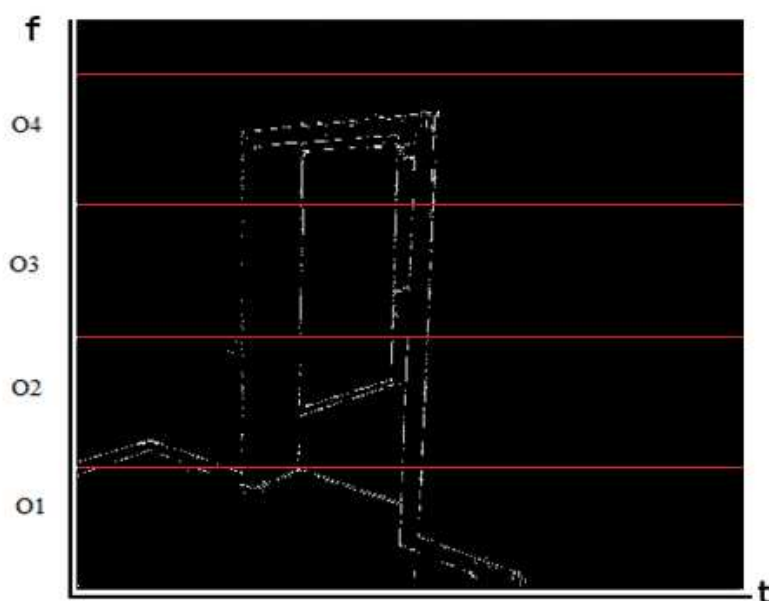


Рисунок 19 – Разбиение на частотные области

Таблица 3 –Таблица частот

Октава\нота	1 нота, Гц	2 нота, Гц	3 нота, Гц	4 нота, Гц	5 нота, Гц
1 область	138.59	155.56	185.0	207.65	233.08
2 область	277.18	311.13	369.99	415.30	466.16
3 область	554.37	622.26	738.99	830.61	932.33
4 область	1108.73	1244.51	1479.98	1661.22	1864.66

Завершающий этап синтеза аудио-сигнала включает суммирование гармоник с соответствующими частотами и их дискретизация с итоговой частотой 1000 Гц, которая была определена экспериментально.

2.2.4 Блок-схема алгоритма преобразования

На рисунке 20 представлена блок-схема алгоритма формирования звукового ряда на основе визуальных данных.



Рисунок 20 – Блок схема алгоритма

2.3 Выводы по главе

В данной главе предлагается алгоритм преобразования визуальных данных, основанный на выделении границ объектов и преобразования их в звуковой ряд, в соответствии с пентатониками первых четырех октав. При помощи этого метода можно сделать функцию, которой можно дополнить существующие системы ориентации в пространстве. Предложенный способ синтеза звуковых сигналов основан на различных частотах с учетом их сочетаемости (консонанса). Данный аспект является важным при длительном прослушивании пользователем аудиоинформации.

3 Разработка мобильного комплекса

В данной главе на основе результатов анализа предметной области существующих систем для слабовидящих и технологий разработки мобильных приложений, были выделены необходимые функции мобильного комплекса:

- определение дальности;
- распознавание образов;
- распознавание речи;
- синтезирование речи;
- преобразование визуальных данных в звук.

А также способы их реализации:

- Arduino nano, HC-05, HC-SR04 (для определения дальности);
- API Vision (для распознавания образов);
- API SpeechKit (для распознавания и синтеза речи);
- метод предложенный во второй главе (для преобразования визуальных данных в звук).

3.1 Требование к системе

Система ориентации в пространстве имеет следующие характеристики:

- языки разработки TS и C++;
- при написании исходного кода использовались Arduino IDE, для работы с контроллером, и Ionic для разработки мобильного приложения;
- для полноценной работы системы необходимо мобильное устройство со встроенной камерой, Bluetooth модулем, микрофоном и динамиком;
- приложение, разработанное на мобильное устройство, является кроссплатформенным, что позволяет не зависеть от операционной системы.

3.2 Описание алгоритма

Алгоритм работы комплекса.

Для начала работы необходимо выбрать функцию. Мобильный комплекс имеет 3 функции: определение дальность, распознавание образа на фото, формирование звукового ряда на основе визуальных данных. Для этого необходимо нажать соответствующую кнопку на экране мобильного устройства или подать голосовую команду.

При выборе функции распознавание образа (рисунок 21), будет сделан снимок с фотокамеры мобильного устройства, который будет отправлен на API Yandex Vision, результатом будет являться ответное сообщение, полученное в виде текстового описания принятого изображения. Затем приложение воспроизведет полученный текст другим API сервисом (SpeechKit).



Рисунок 21 – Блок-схема распознавания образа

Голосовое управление комплексом (рисунок 22). Приложение получает предварительную команду, затем производит аудио запись. Далее используется API Speechkit, которое возвращает распознанный текст, в соответствии с полученным сообщением выполняется или не выполняется команда.

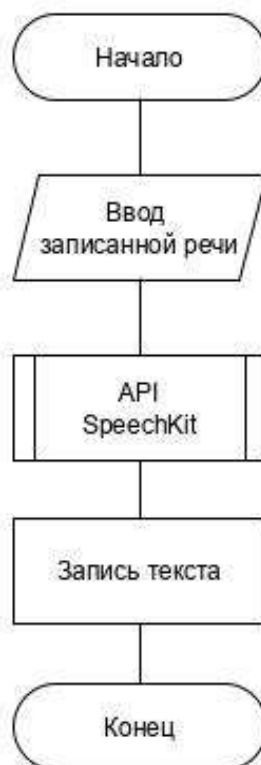


Рисунок 22 – Распознавание речи



Рисунок 23 – Синтез речи

Для определения дальности с приложения будет послан управляющий сигнал на микроконтроллер (посредством беспроводной связи Bluetooth), который возвратит действительное значение дальность до объекта, приложение мобильного устройства в свою очередь оповестит пользователя соответствующим звуковым сигналом.

При выборе последней функции, приложение преобразует полученное изображение с фотокамеры мобильного устройства, выделит границы объектов и преобразует их в звуковой ряд.

3.3 Структурная схема

Структурная схема мобильного комплекса представлена на рисунке 24.

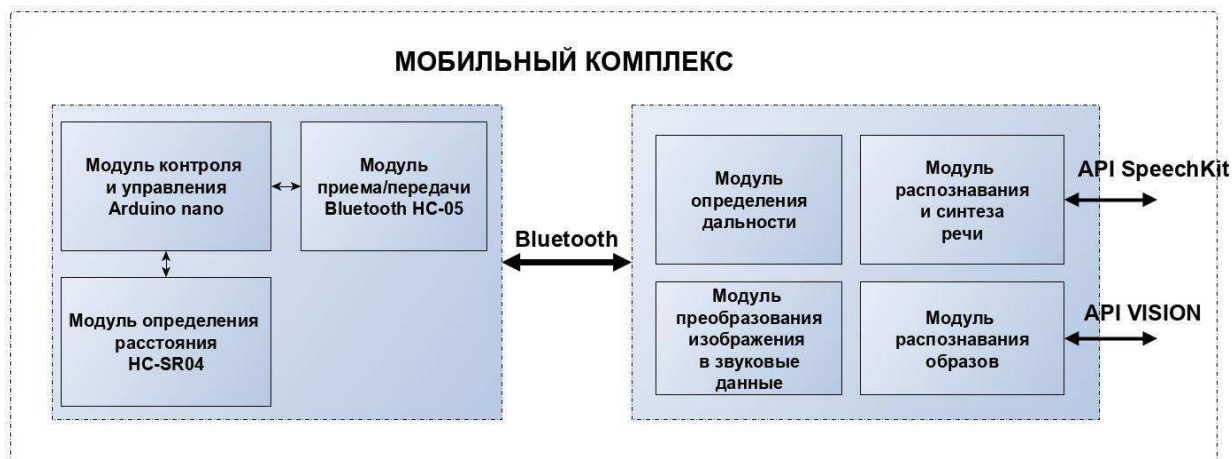


Рисунок 24 – Структурная схема мобильного комплекса

3.3.1 Определение дальности

Модуль контроля и управления. В качестве блока контроля и управления используется микроконтроллер Arduino nano. Блок управления получает управляющий сигнал с мобильного устройства посредством Bluetooth модуля HC-05, для определения дальности при помощи CH-SR04. Затем вычисляет значение и формирует ответ мобильному устройству.

Прием и передача. В качестве блока приёма и передачи выступает Bluetooth модуль HC-05. Bluetooth модуль получает информацию от микроконтроллера и выполняет все необходимые действия, для передачи информации на мобильный телефон. принимает управляющие сигналы от мобильного телефона и передаёт их на микроконтроллер.

Исполнительные устройства. Может представлять собой мобильные телефоны(ноутбуки) имеющие Bluetooth передатчик. Мобильный телефон

принимает, обрабатывает и представляет в удобном пользователю виде полученную от Bluetooth модуля информацию.

Описание алгоритма. Arduino генерирует высокий сигнал на выводе TRIG длиной 10 микросекунд, вслед за этим, дальномер выпустит 8 импульсов и поднимет высокий уровень на выводе ECHO, перейдя в режим ожидания отраженного сигнала. В это время микроконтроллер считает длину сигнала, как только на выводе ECHO будет низкий уровень, то произойдет расчет расстояния и передача по его на Bluetooth модуль, который в свою очередь передаст значение на мобильное приложение.



Рисунок 25 – Блок-схема определение дальности

3.3.2 Описание сервиса Yandex Vision

API (Application Programming Interface) – это набор готовых инструментов какого-либо приложения, которые можно использовать в сторонних продуктах.

Yandex Vision – сервис компьютерного зрения для анализа изображений. Сервис компьютерного зрения Yandex Vision анализирует переданное изображение и возвращает результат.

Возможности:

- распознавание текста;
- обнаружение лиц;
- классификация изображений.

3.3.3 Описание сервиса Yandex SpeechKit

SpeechKit – технология распознавания речи и синтеза речи от российской компании Яндекс. Сервис позволяет переводить речь в текст и обратно.

Возможности:

- распознавание речи;
- синтез речи;
- выделение смысловых объектов речи.

Описание параметров в теле запроса генерации и распознавания речи представлены в таблице 4 и в таблице 5 соответственно.

Таблица 4 – Параметры запроса синтеза речи

Параметр	Значение	Описание
text	Текст в UTF-8	Обязательный параметр

lang	ru-RU	Язык
voice	oksana	Голос синтезированной речи
speed	0.1 – 3.0	Скорость синтезированной речи
format	lpc	Формат синтезированного сигнала
sampler	48кГц	Частота дискретизации

Таблица 5 – Параметры запроса распознавания речи

Параметр	Значение	Описание
lang	ru-RU	Язык
topic	general	Языковая модель
profanityFilter	true/false	Фильтр ненормативной лексики
format	lpc	Формат синтезированного сигнала
sampler	48 кГц	Частота дискретизации

3.3.4 Интерфейс приложения

На рисунке 26 представлен интерфейс приложения.

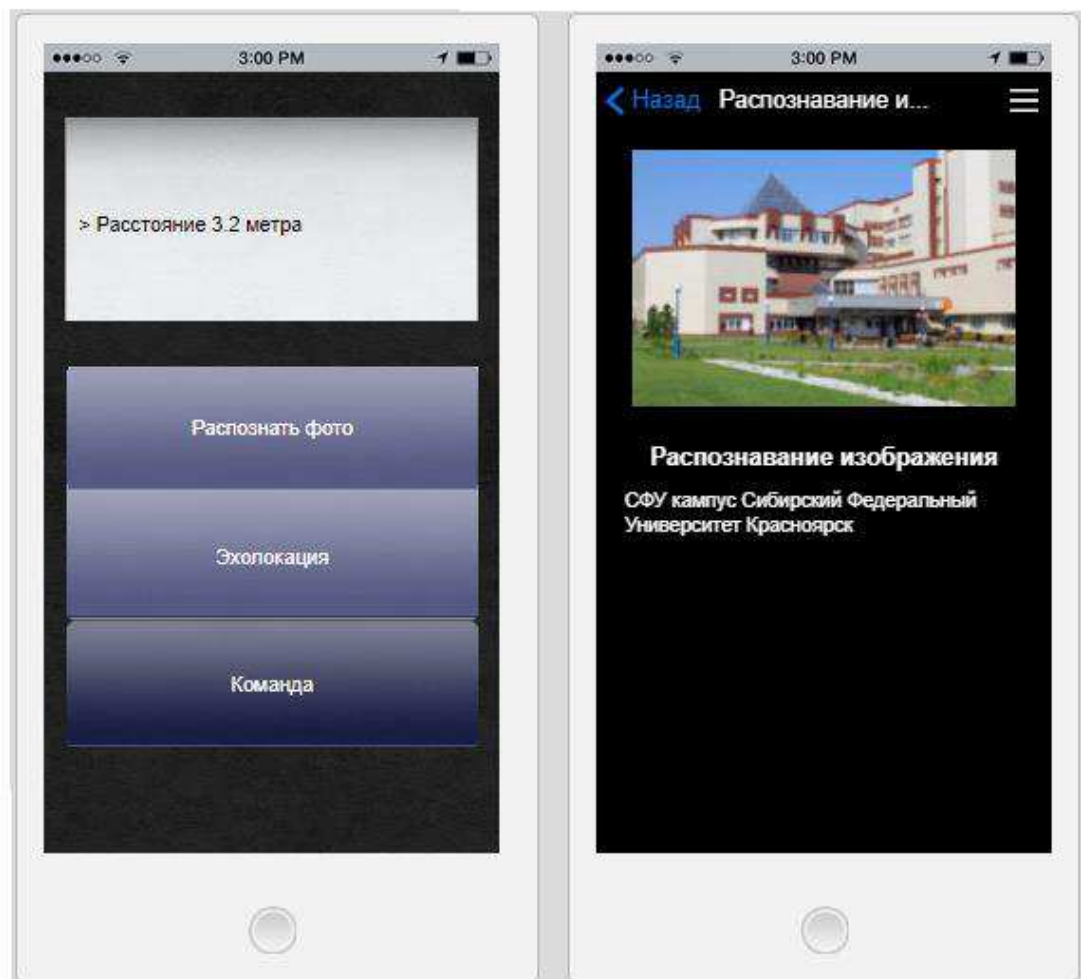


Рисунок 26 – Интерфейс приложения

Управление комплексом возможно при помощи графического интерфейса или посредством голосовых команд. Arduino с помощью ультразвукового дальномера, обнаружит препятствие и передаст значение в приложение мобильного телефона, которое в свою очередь оповестит пользователя.

Камера фиксирует изображение и передает его в Vision, далее устройство определяет на изображении предметы, текст, образы. Затем информирует пользователя в виде звукового сигнала на русском языке.

Также в комплексе есть функция, которая позволит пользователю слышать звуковой сигнал, сформированный на основе изображения с камеры.

Приложение может работать в постоянном режиме или же выполнить какую-либо функцию по требованию пользователя.

ЗАКЛЮЧЕНИЕ

В результате данной работы по разработки и проектированию мобильного комплекса для слабовидящих, были решены следующие задачи:

- изучены основные системы для ориентации в пространстве для слабовидящих;
- разработана структурная схема мобильного комплекса;
- разработаны основные алгоритмы системы;
- разработан алгоритм преобразования изображения в звук
- реализована программа взаимодействия Arduino и мобильного телефона;
- разработан прототип мобильного приложения.

И предложен метод синтеза аудио-сигналов на основе визуальных данных, в основу которого был положен принцип формирования звукоряда на основе пентатоники.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Эхолокация в области машинного зрения [Электронный ресурс] – Режим доступа: <https://core.ac.uk/download/pdf/53080988.pdf>
2. Описание The Voice [Электронный ресурс] – Режим доступа: <https://www.seeingwithsound.com/manualru/ThevOICeTrainingManualru.htm>
3. Википедия – свободная интернет-энциклопедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/>
4. HTML [Электронный ресурс] – Режим доступа: <http://html5book.ru/osnovy-html/>
5. CSS [Электронный ресурс] – Режим доступа: <http://html5book.ru/osnovy-css/>
6. JavaScript [Электронный ресурс] – Режим доступа: <http://html5book.ru/osnovy-javascript/>
7. Тарасьев, А.А. Основные методы цифровой обработки изображений [Электронный ресурс] / А.А. Тарасьев // Территория науки. – 2016. – №3. – Режим доступа: <https://cyberleninka.ru/article/v/realizatsiya-osnovnyh-metodov-tsifrovoy-obrabotki-izobrazheniy-na-yazyke-s>
8. Белим, С.В. Сравнение алгоритмов выделения контуров [Электронный ресурс] / С.В. Белим // Компьютерная оптика. – 2015. – №1. – Режим доступа: <https://cyberleninka.ru/article/n/sravnenie-algoritmov>
9. Теория музыки [Электронный ресурс] – Режим доступа: <http://music-theory/>
10. Адибаев, Б.М. Влияние звуковых волн на организм [Электронный ресурс] / Б.М. Адибаев // Гигиена и экология. – 2018. – №1. – Режим доступа: <https://cyberleninka.ru/article/v/vliyanie-zvukovyh-voln-na-organizm>
11. Операторы Собела и Лапласа [Электронный ресурс] – Режим доступа: <http://robocraft.ru/blog/computervision/460.html>
12. Методы выделения объектов [Электронный ресурс] – Режим доступа: <https://moluch.ru/archive/193/48447/>

13. Оптимальные алгоритмы выделения контуров [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/v/optimalnye-algoritmy-vydeleniya-konturov-izobrazheniya-v-sisteme-tehnicheskogo-zreniya>
14. Ionic [Электронный ресурс] – Режим доступа: <https://www.ionic.com/>
15. React native [Электронный ресурс] – Режим доступа: <https://dou.ua/lenta/articles/react-vs-real/>
16. Детектор границ Канни [Электронный ресурс] – Режим доступа: <http://elib.spbstu.ru/dl/2/v17-3714.pdf/v17-3714.pdf>
17. Выделение границ ландшафтов [Электронный ресурс] – Режим доступа: <http://www.vestnik.vsu.ru/pdf/geograph/2013/02/2013-02-04.pdf>
18. Компьютерное зрение [Электронный ресурс] – Режим доступа: http://ermak.cs.nstu.ru/neurotech/html/metodmat/ist2016/Lec_14.pdf
19. Apache Cordova [Электронный ресурс] – Режим доступа: <https://cordova.apache.org/>
20. Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular [Электронный ресурс] – Режим доступа: <https://ionicframework.com/>

