|  |  |
| --- | --- |
| Gerb-BMSTU_01 | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение  высшего образования  «Московский государственный технический университет  имени Н.Э. Баумана  (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «Робототехника и комплексная автоматизация»  КАФЕДРА «Теория механизмов и машин»  ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ  Студент РК2-41М О.В. Зудина  (Подпись, дата)  Руководитель С.А. Воротников  (Подпись, дата)  Консультант Д.В. Сащенко  (Подпись, дата) | |
| 2020 г. | |

РЕФЕРАТ

Отчет по преддипломной практике содержит 24 страницы машинописного текста, 10 рисунков, 1 таблицу, 6 источников.

ПЛОСКИЙ РЫЧАЖНЫЙ МЕХАНИЗМ, КИНЕМАТИЧЕСКИЕ ХАРАКТЕРИСТИКИ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, ОБРАБОТКА ВИДЕОПОТОКА, РАСПОЗНАВАНИЕ ИЗОБРАЖЕНИЙ.

В рамках преддипломной практики была поставлена цель разработки мобильного приложения для экспериментального определения параметров кинематики плоских рычажных механизмов и их моделей с использованием компьютерного зрения. Объектом исследования является обработка видеопотока полного цикла работы механизма.

Практическая ценность: данное приложение можно использовать для разработки современной лабораторной работы кафедры РК2.

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ 5](#_Toc42433279)

[ВВЕДЕНИЕ 7](#_Toc42433280)

[1 Разработка мобильного приложения 8](#_Toc42433281)

[1.1 Функциональная схема 8](#_Toc42433282)

[1.2 Компьютерное зрение 9](#_Toc42433283)

[1.3 Алгоритмы распознавания изображений 10](#_Toc42433284)

[1.4 Алгоритмы фильтрации изображений 10](#_Toc42433285)

[1.4.1 Бинаризация по порогу 11](#_Toc42433286)

[1.4.2 Быстрое преобразование Фурье, ФНЧ, ФВЧ 12](#_Toc42433287)

[1.4.3 Вейвлет-преобразование 13](#_Toc42433288)

[1.4.4 Корреляция 14](#_Toc42433289)

[1.4.5 Фильтрация контуров 14](#_Toc42433290)

[1.5 Алгоритмы обработки результатов фильтрации 15](#_Toc42433291)

[1.5.1 Контурный анализ 15](#_Toc42433292)

[1.5.2 Особые точки 15](#_Toc42433293)

[1.6 Библиотеки для компьютерного зрения 16](#_Toc42433294)

[1.6.1 OpenCV 16](#_Toc42433295)

[1.6.2 CCV 17](#_Toc42433296)

[1.6.3 Tensorflow 18](#_Toc42433297)

[1.6.4 Сравнение библиотек 18](#_Toc42433298)

[1.7 Мобильное приложение 19](#_Toc42433299)

[ЗАКЛЮЧЕНИЕ 22](#_Toc42433300)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc42433301)

[СПИСОК ИСПОЛЬЗУЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 24](#_Toc42433302)

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ

Траектория — множество точек, представляющее путь движения тела или точки.

Кинематические характеристики — мера положения, отображение формы и характера механических движений объекта.

Кинематический анализ — определение кинематических характеристик механизма.

Аналог линейной скорости — первая производная линейной функции положения точки по обобщенной координате.

Аналог ускорения — вторая производная линейной функции положения точки по обобщенной координате.

Мобильное устройство — любое небольшое устройство, которое содержит дисплей и клавиатуру.

Видеозапись — электронная технология записи на физической носитель визуальной информации, представленной в форме видеосигнала или цифрового потока видеоданных.

Компьютерное зрение — научное направление в области искусственного интеллекта и связанные с ним технологии получения изображений объектов реального мира, их обработки и использования полученных данных для решения различных прикладных задач без полного или частичного участия человека.

Машинное зрение — применение компьютерного зрения в задачах робототехники, промышленности и производства.

Пиксель — pixel (pictures element) — наименьший логический элемент двухмерного цифрового изображения в растровой графике или элемент матрицы дисплеев, формирующих изображение.

RGB-изображение — red, green, blue — цветовая модель, описывающая способ кодирования цвета с помощью трех цветов.

HSV-изображение — hue, saturation, value — цветовая модель, в которой координатами цвета являются цветовой тон, насыщенность и яркость.

Дедупликация — метод сжатия массива данных, использующий в качестве алгоритма сжатия исключение дублирующих копий повторяющихся данных.

ВВЕДЕНИЕ

Практические занятия по дисциплине «Теория механизмов и машин» кафедры РК2 подразумевают под собой закрепление студентами теоретических основ. Это важный этап в становлении на путь специалиста в этой области [1].

Лабораторный практикум [2] данной дисциплины сформировался в нынешнем виде более 30 лет назад. На данный момент эти занятия проводят на классических, но довольно старых стендах с оборудованием. Этот факт в современном мире доставляет ряд неудобств, таких как: ограниченная пропускная способность каждого стенда, большие занимаемые объемы этим оборудованием, большая стоимость каждого вида оборудования, что влечет за собой повышенную осторожность при работе с ним.

Одним из возможных решений этой задачи может стать внедрение современных технологий в обучающие и контролирующие знания процессы. Например, замена громоздких стендов компьютерными программами, с которыми студент сможет работать на собственной технике (персональный компьютер, мобильное устройство) или кафедральном аналоге. Данное решение сможет облегчить и ускорить работу как обучающим, так и контролирующим преподавателям, потому что в этом случае студент может не контактировать с оборудованием напрямую.

Разработка мобильного приложения

В рамках преддипломной практики было принято решение об использовании компьютерного зрения для решения поставленных задач, проведен обзор библиотек машинного зрения, разработано мобильное приложение, позволяющее фиксировать траекторию маркера.

Функциональная схема

В состав стенда разрабатываемой лабораторной работы входит мобильное устройство с предустановленным приложением для фиксации и обработки траектории маркера. На рисунке 1 представлена функциональная схема мобильного приложения в виде дерева функций в нотации ARIS.

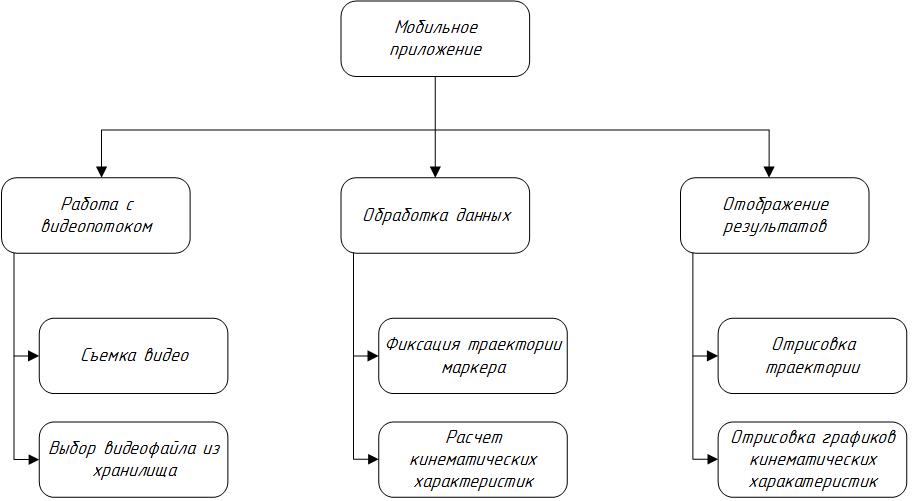


Рисунок 1. Дерево функций мобильного приложения

Описываемое приложение должно содержать несколько модулей:

* модуль компьютерного зрения;
* модуль обработки данных;
* модуль отображения графиков.

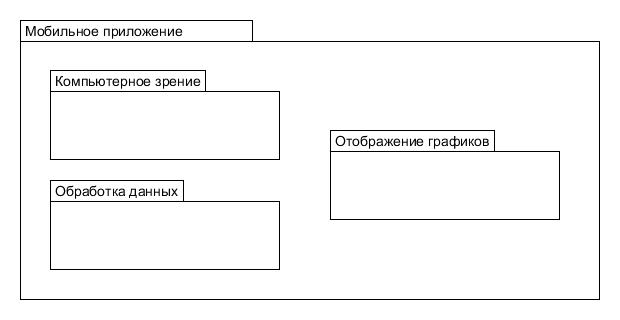


Рисунок . Диаграмма пакетов мобильного приложения

Компьютерное зрение

Компьютерное зрение является активно развивающимся научным направлением в области искусственного интеллекта. Также включает технологии получения изображений объектов реального мира, их обработки и использования полученных данных для решения различных прикладных задач без полного или частичного участия человека.

Компьютерное зрение включает в себя несколько разделов:

* Обработка изображений (image processing, на входе и выходе изображения);
* Анализ изображений (image analysis, работа с 2D-изображениями);
* Распознавание образов (pattern recognition, распознавание, обучение на абстрактных числовых величинах, полученных в том числе и из изображений);
* Компьютерное зрение (computer vision, восстановление 3D-структуры по 2D-изображениям, принятие решений о физических объектах, основываясь на их изображениях);
* Фотограмметрия (photogrammetry, измерение расстояний между объектами по 2D-изображениям);
* Машинное зрение (machine vision, решение промышленных и производственных задач).

Таким образом, с помощью компьютерного зрения можно решить задачу построения траекторий некоторых заданных точек плоского рычажного механизма по видеопотоку полного цикла работы. Применение данного направления позволит оптимизировать процесс получения множества точек траекторий и автоматизировать процесс расчета кинематических характеристик механизмов.

## Алгоритмы распознавания изображений

В данной работе был рассмотрен один из самых часто применяемых способов распознавания изображений, состоящий из двух этапов. Первый этап заключается в фильтрации графического файла. Под фильтрацией понимают выделение областей, которые могут быть интересны для анализатора объектов. Обычно алгоритмы фильтрации представляют собой применение единого преобразования на все точки изображения. Анализ и обработка полученных на первом этапе данных представляют собой второй этап распознавания. Результатами второго этапа являются свойства объектов или сами объекты, графические файлы которых были представлены в виде исходных данных

Алгоритмы фильтрации изображений

Ниже рассмотрены некоторые методы выделения областей на изображениях. Данные методы применяют единое математическое преобразование ко всем точкам изображения. Точкой может являться один пиксель или группа пикселей. В качестве входных данных алгоритмов фильтрации являются гистограммы изображений (рис. 3).



Рисунок . Гистограмма изображения

Гистограмма отображает количество пикселей изображения, соответствующих определенному уровню яркости.

На этапе фильтрации анализ областей не производится, но точки, прошедшие фильтрацию, являются входными данными для этапа анализа и распознавания объектов.

Бинаризация по порогу

Бинаризация является одним из самых простых способов преобразования изображения. В случае, когда входными данными для данного метода являются изображения в градациях серого или RGB-изображения, порогом бинаризации является значение цвета. Выбор порога может происходить адаптивно, что повышает точность выделения областей с особыми характеристиками. Адаптивный алгоритм может представлять собой выбор математического ожидания, моды или максимального пика на гистограмме.

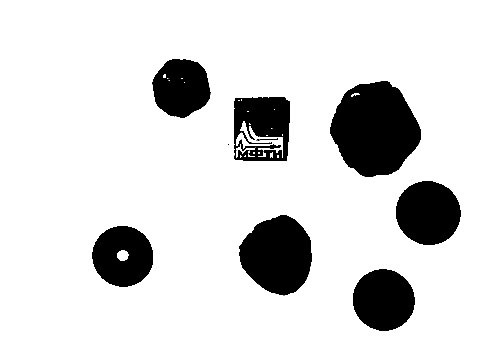


Рисунок 4. Пример работы бинаризации по порогу

При рассмотрении HSV-изображения бинаризация позволяет обнаруживать метки заданного цвета, данный подход используется во многих известных инструментах распознавания изображений.

Быстрое преобразование Фурье, ФНЧ, ФВЧ

Быстрое преобразование Фурье[3] является одним из классических методов обработки сигналов. Широко применяется одномерное преобразование для сжатия звуковых файлов и изображений. В общем случае преобразование Фурье представляет операцию, сопоставляющую одной непериодической функции вещественной переменной другую функцию вещественной переменной, являющуюся периодической.

Для анализа изображения используют ресурсоемкое двумерное преобразование, которое обычно заменяют сверткой интересующей области с применением ФВЧ (фильтра высоких частот) или ФНЧ (фильтра низких частот). Для каждой точки изображения выбирается диапазон и перемножается с фильтром того же размера, результатом такой свертки является новое значение точки.

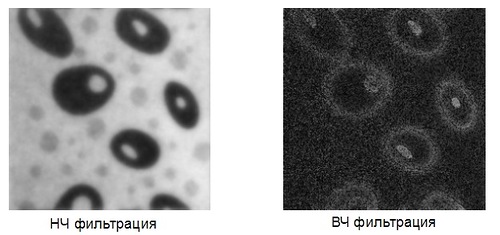
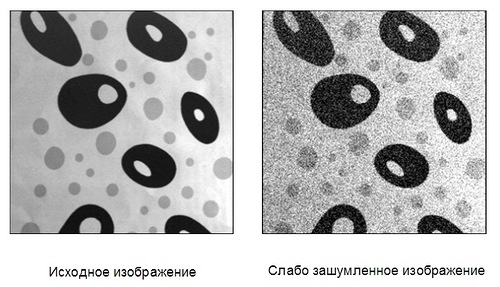
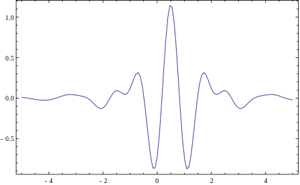
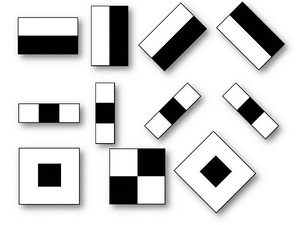


Рисунок 5. Применение ФНЧ и ФВЧ

Вейвлет-преобразование

Вейвлет-преобразование представляет собой свертку с использованием произвольной характеристической функции. Название «вейвлет» происходит от английского «wavelet» — небольшая волна, рябь.



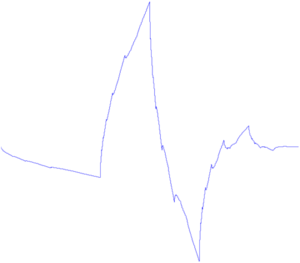
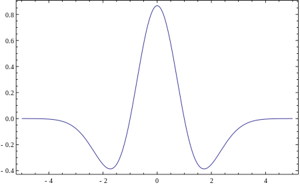


Рисунок 6. Классические функции вейвлет-преобразования

Во многих командах вейвлет-анализом называют поиск произвольного повторяющегося элемента на изображении при помощи свертки с моделью этого элемента. К классическим функциям вейвлет-преобразования относят трехмерный вейвлет Хаара, и двухмерные вейвлеты Мейера, Мексиканская шляпа и Дебоши. Данные функции используются для классификации и сжатия изображений.

Корреляция

Корреляция лежит в основе вейвлет-преобразования и применяется к видеопотокам для нахождения сдвигов. Изображения накладываются друг на друга, и в случае отсутствия корреляции делают вывод о движении кадра или объекта в кадре.

Фильтрация контуров

Существует отдельный класс фильтров, предназначенный для фильтрации границ и контуров и применяемый для достаточно сложных, но хорошо выделяемых объектов на изображении. Задачу фильтрации контуров решают с помощью следующих алгоритмов:

* оператор Кэнни;
* оператор Собеля;
* оператор Лапласа;
* оператор Прюитт;
* оператор Робертса.

На рисунке 7 приведен пример использования оператора Кэнни для распознавания контуров оборудования.

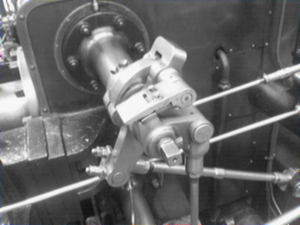
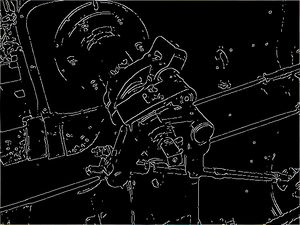
 

Рисунок 7. Применение оператора Кэнни

Алгоритмы обработки результатов фильтрации

Результаты фильтрации являются входными данными для алгоритмов логической обработки. Данные алгоритмы позволяют перейти от изображения к объектам или их свойствам.

Контурный анализ

Полученные границы после применения алгоритмов фильтрации достаточно просто преобразуются в контуры. Например, для оператора Кэнни это происходит автоматически, для остальных операторов требуется дополнительная бинаризация. Получить контур можно с помощью алгоритма «жука», закодировав его впоследствии с помощью цепного кода Фримена.

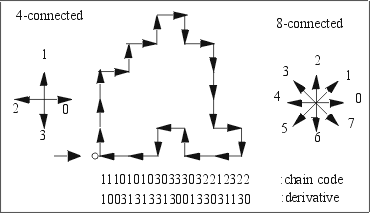
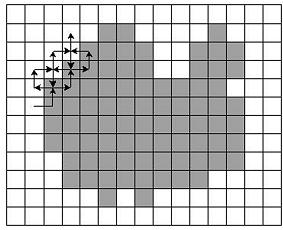


Рисунок 8. Схема работы алгоритма «жука» и пример цепного кода Фримена

К сожалению, данный способ редко применяется в реальных задачах. Неудачный результат поиска границы или слишком высокий уровень шумов не позволяют использовать контурный анализ.

Особые точки

Особыми точками называют уникальные характеристики объекта, позволяющие сопоставить объект сам с собой или с похожими классами объектов. Существует огромное разнообразие алгоритмов, выделяющих особые точки. Данные алгоритмы делятся на несколько групп по классу особых точек:

* Поиск точек с течением времени;
* Поиск точек при смене освещения и положения;
* Поиск стабильных точек.

Первый класс предназначен для поиска объекта между некоторыми кадрами видеопотока, второй класс используется для обучения и классификации типов объектов, третий класс позволяет находить особые точки даже при повороте изображения, но алгоритмы данного класса являются самыми долгими и затратными.

Библиотеки для компьютерного зрения

### OpenCV

Open Source Computer Vision Library[4] является библиотекой компьютерного зрения с открытым исходным кодом. Содержит реализации множества алгоритмов обработки изображений, численных алгоритмов и алгоритмов компьютерного зрения. Основными языками программирования данной разработки являются C/C++[5], но существует большое количество инструментов, позволяющих интегрировать OpenCV с различными языками программирования: Java, Python[6], Ruby, Matlab и других. Также библиотека поддерживает большое количество платформ: Microsoft Windows, Linux, Mac OS, Android, iOS.

OpenCV содержит несколько модулей, каждый из которых отвечает за определенную задачу:

* Модуль основной функциональности, включает базовые структуры, вычисления и линейную алгебру;
* Обработка изображений, фильтрация, преобразования цветов;
* Модуль ввода и вывода изображений и видео;
* Модели машинного обучения;
* Распознавание плоских примитивов;
* Обнаружение объектов на изображении;
* Анализ движения и отслеживание объектов;
* Модуль поиска ближайших соседей;
* Модуль калибровки камеры, обработка трехмерных данных;
* Модуль ускорения с применением CUDA.

CCV

Библиотека ccv[7] написана на языке C и позиционируется как легкая и компактная альтернатива OpenCV, реализующая только основные алгоритмы распознавания изображений. Данная библиотека предназначена не для экспериментов с разными алгоритмами, а для практического использования в конкретных прикладных задачах.

В библиотеке ccv представлены следующие алгоритмы:

* Быстрый алгоритм распознавания неподвижных объектов, основанный на пороге яркости;
* Алгоритм «Хищник»;
* Алгоритм распознавания текста, основанный на преобразовании ширины линий;
* Алгоритм обнаружения наборов точек;
* Алгоритм распознавания сложных деформируемых объектов;
* Алгоритм каскадного обнаружения устойчивых признаков объектов;
* Обнаружение объектов в видеопотоке.

Данная библиотека поддерживает использование REST-API с использованием протокола HTTP, использует прозрачный кэш для дедупликации данных, присутствует возможность классификации данных с применением глубоких сверточных нейронных сетей.

Tensorflow

Библиотека Tensorflow[8] используется для машинного зрения, но не позволяет распознавать объекты на изображении, а применяется на этапе обучения для построения и тренировки нейронной сети, решает задачу классификации образов. Реализована на языках C++ и Python, с 2015 является библиотекой с открытым исходных кодом.

Сравнение библиотек

В данном разделе представлено сравнение библиотек для компьютерного зрения, рассмотренных выше.

Таблица 1. Сравнение библиотек компьютерного зрения

|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека для компьютерного зрения | OpenCV | ccv | Tensorflow |
| Возможность распознавания маркеров стандартными методами | + | + | - |
| Возможность распознавания нескольких маркеров на одном изображении | + | + | - |
| Возможность распознавания объектов в видеопотоке | + | + | - |
| Язык программирования, на котором реализована библиотека | C/C++ | C | C++, Python |
| Открытый исходный код | + | + | + |
| Возможность интеграции с операционной системой Android | + | - | + |
| Версия последнего обновления | 4.1.2 | 0.7 | 2.1.0 |
| Дата последнего обновления | 11.12.2019 | 23.12.2014 | 11.12.2019 |

Мобильное приложение

На основе дерева функций была составлена диаграмма активностей мобильного приложения, представленная на рисунке 9.

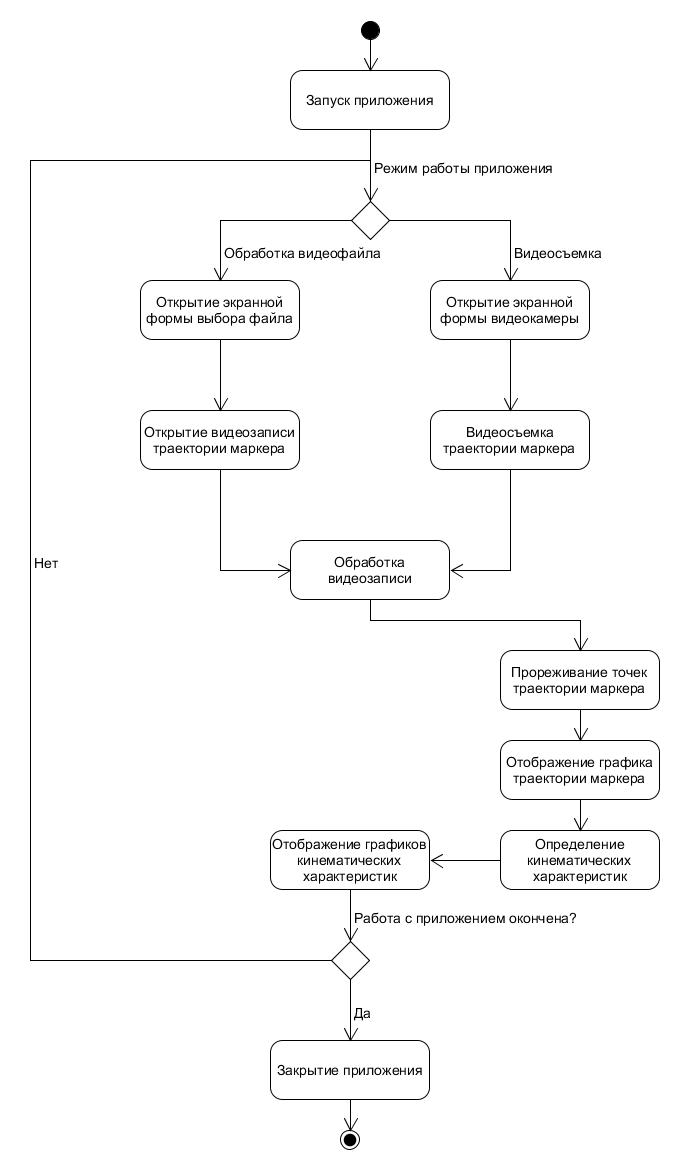


Рисунок 9. Диаграмма активности приложения

Разработка приложения велась на языке программирования Java[9], поддерживаемым мобильной платформой Android. Используемый язык программирования является строго типизированным и поддерживает объектно-ориентированную парадигму.

В качестве среды разработки была использована Android Studio[10] версии 3.6. В процессе разработки модуля компьютерного зрения в проект были включены зависимости пакета openCVLibrary343, то есть, реализация библиотеки OpenCV для платформы Android.

В архитектуре приложения был использован шаблон Model-View-ModelView (MVVM). Концепция данного шаблона заключается в отделении логики представления данных от логики работы приложения:

* *Model* – логика, связанная с данными приложения;
* *View* – экранная форма экрана, в которой располагаются все необходимые элементы для отображения информации;
* *ViewModel* – объект, в котором описывается логика поведения *View* в зависимости от результата работы *Model*.

В рамках данной работы были разработаны следующие классы:

* *MainActivity* – главный компонент приложения, управляющий экранными формами и предоставляющий взаимодействие с пользователем;
* *StartFragment* – начальная экранная форма, позволяющая выбрать режим работы с видеокамерой или файлом из хранилища;
* *FileFragment* – экранная форма для работы с файлом из хранилища;
* *CameraFragment* – экранная форма для работы с видеокамерой;
* *ChartFragment* – экранная форма для подготовки и прореживания данных и последующего отображения данных на графиках;
* *KinematicHandler* – модуль определения кинематических характеристик.

На рисунке 10 представлена диаграмма классов разрабатываемого приложения.

|  |
| --- |
| C:\Users\Каито\Documents\Umlet\works\class.jpg  Рисунок 10. Диаграмма классов мобильного приложения |

ЗАКЛЮЧЕНИЕ

В рамках данной преддипломной практики были получены следующие результаты:

* принято решение об использовании компьютерного зрения;
* рассмотрены основные алгоритмы распознавания изображений;
* произведено сравнение различных библиотек компьютерного зрения и выбор основных инструментов для дальнейшей разработки;
* разработано мобильное приложение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Теория механизмов и машин. Под ред. К. В. Фролова. – М.: Высшая школа, 1987.
2. Тарабарин В. Б., Кузенков В. В., Фурсяк Ф. И. Лабораторный практикум по теории механизмов и машин. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 80 с., ил.
3. Е.А. Власова. Ряды. Математика в техническом университете. – М.: Издательство МГТУ им. Н.Э. Баумана, 2006.
4. OpenCV Documentation [Электронный ресурс]. – 2019. – URL: http://docs.opencv.org/ (дата обращения: 23.12.2019).
5. C++ Documentation [Электронный ресурс]. – 2019. – URL: https://docs.microsoft.com/ru-ru/cpp (дата обращения: 23.12.2019).
6. Python3 Documentation [Электронный ресурс]. – 2019. – URL: https://docs.python.org/3/ (дата обращения: 23.12.2019).
7. ccv Documentation [Электронный ресурс]. – 2014. – URL: http://libccv.org/doc/ (дата обращения: 23.12.2019).
8. Tensorflow Documentation [Электронный ресурс]. – 2019. – URL: https://www.tensorflow.org/versions (дата обращения: 23.12.2019).
9. Java Documentation [Электронный ресурс]. – 2020. – URL: https://docs.oracle.com/en/java/ (дата обращения: 07.06.2020).
10. Android Studio Documentation [Электронный ресурс]. – 2020. – URL: https://developer.android.com/docs (дата обращения: 07.06.2020).

# СПИСОК ИСПОЛЬЗУЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Microsoft® Office Word 2010
2. Mathcad 15
3. MATLAB R2019b
4. CLion 2019.2.2
5. OpenCV 4.1.2
6. ccv 0.7
7. Tensorflow 2.1.0
8. Android Studio 3.6
9. Umlet 14.3.0