МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра: Информатики и автоматизации научных исследований**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

Профиль подготовки: «Инженерия программного обеспечения»

**ОТЧЕТ**

**Тема:**

**«Методы контроля движения в задаче навигации в закрытых помещениях»**

Допущена к защите Выполнил:

Заведующий кафедрой: студент группы 381306-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Андреев В. С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись

Научный руководитель:

к.т.н., доцент Осипов М.П.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ученая степень, ученое звание,

ф.и.о.

Нижний Новгород  
2018

1. Введение

Внутренняя структура аэропортов, вокзалов, торговых и развлекательных центров имеют сложную конфигурацию. Современные здания зачастую представляют собой большой комплекс из этажей с лестницами, коридорами и комнатами. Ориентироваться в таких зданиях становиться достаточно трудной задачей. Использование доступной технологии спутниковых систем навигации типа GPS-GLONASS внутри зданий крайне затруднено, поскольку бетонные стены и перекрытия сводят уровень сигнала к нулю. Традиционным решением навигации в таких зданиях являются визуальные ориентиры на стенах помещений. Недостаток такого подхода состоит в большой информационной нагрузке на человека. Технологичные компании по всему миру занимаются задачей автоматизации поиска местоположения пользователя в закрытых помещениях. Существуют разные подходы к решению этой задачи, основанные на использовании: различных сетей передачи данных (Bluetooth, Wi-Fi, FM-радиоволны, UWB); магнитного поля Земли; RFID-меток, QR-кодов, MEMS сенсоров, и т.п. [1]. Однако такие подходы имеют ограничение по точности и надежности, либо трудности в установке соответствующей инфраструктуры и её эксплуатации, либо требуют существенных финансовых вложений.

1. Формулировка проблемы

Цель работы состоит в развитии подхода автономной инерциальной навигации, основанного на использовании микромеханических инерционных датчиков и информации о планировке здания. Такой подход является самым “дешевым” из существующих технологий навигации, однако показатели точности позиционирования невысоки. В работе предложены алгоритмы обеспечивающие повышение точности и надежности позиционирования.

Декомпозиция предложенной цели в конкретные задачи:

1. Контроль движения с использованием поэтажного плана здания
2. Разработка программного обеспечения надежного распознавания объектов из видео потока для коррекции местоположения пользователя в задачах навигации в закрытых помещениях (выбор из видеопотока наиболее подходящего изображения предообработка изображения)
3. Адаптация для мобильных платформ и последующая оптимизация алгоритмов распознавания объектов из видеопотока (использовать различные средства аппаратного ускорения, доступные на мобильном устройстве)
4. Разработка программного обеспечения клиент-серверной системы навигации в закрытых помещениях на современных мобильных устройствах
5. Контроль движения с использованием поэтажного плана здания

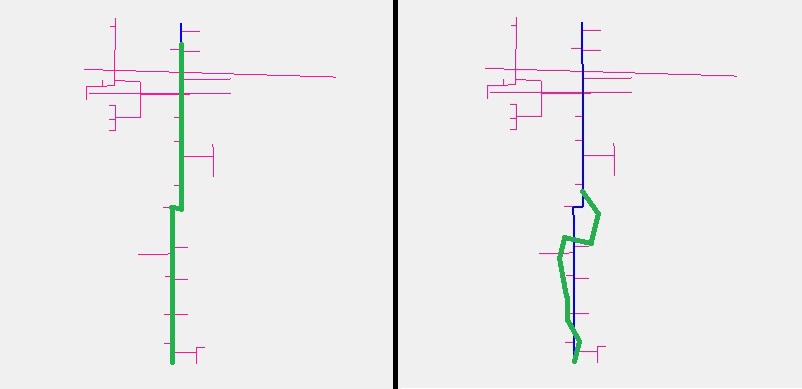


Рисунок 1Результат работы алгоритма инерциальной навигации с контролем движения и без него

В ходе движения могут возникнуть ситуации ложного поворота вследствие несовершенства показаний микромеханических инерционных датчиков. В этом случае, путь движения пользователя может существенно отклониться. Для решения этой проблемы предлагается при достижении пользователем поворота (разветвления) в конфигурации поэтажного плана помимо основного пути движения одновременно вести альтернативные. Каждый альтернативный путь движения имеет параметр приоритета, который будет уменьшаться в ходе дальнейшего сопровождения по маршруту, если вычисленная предполагаемая позиция пользователя по показаниям инерционных датчиков не будут соответствовать конфигурации поэтажного плана. Пути с наименьшим приоритетом удаляются. А путь с наибольшим параметром приоритета становится текущим.

Алгоритм приоритезации маршрутов движения основан на двух принципах:

1. Принцип совпадения направлений;
2. Принцип совпадения конфигураций;

Принцип совпадения направлений – если текущий путь истинный, то текущий вектор пути должен быть сонаправлен с текущим вектором, полученным на основе автономной навигации.

Принцип совпадения конфигураций – если текущий шаг, совершенный пользователем, не принадлежит никакому отрезку выбранного пути, то выбранный маршрут считается не верным. Так как точки пространства, не принадлежащие отрезкам пути, принадлежат различным физическим барьерам в процессе движения (стены).

Рассмотрим, каким образом алгоритм эксплуатирует приведенные принципы. Алгоритм построен на идеи случайного процесса, а именно на случайных цепях Маркова. Каждому маршруту движения приписывается номер. И каждому такому номеру сопоставляется вероятность, которая показывает что в конкретный момент времени t, пользователь находится на текущем отрезке (или на текущем пути движения), вероятности выбирается таким образом, что бы выполнялась аксиома нормировки для случайной величины, а именно сумма вероятностей для каждого номера ровна 1. Рассуждая подобным образом, получаем, что в каждый момент времени t, систему можно описать как дискретную случайную величину. Стоит заметить, что максимальное число маршрутовM, которые рассматриваются на любой момент времени tравно шести, цифра была, выбрана основываясь, на двух эвристиках:

1. В конфигурации любого здания преобладают перекрестки, которые делят маршрут движения на два.
2. Различие в конфигурация любых двух маршрутов возможно отследить уже после первого перекрестка. Так как длина коридоров сильно отличается друг от друга.

Алгоритм определения типа движения

Типичная конфигурация зданий - это наличие лестничных пролетов, сосредоточенные в некоторых частях здания, соответственно, если будет возможность определить, что пользователь системы находиться на лестницы, эта информация позволит сократить возможные места нахождения пользователя, и скорректировать положение пользователя в приложении. То есть определение типа движения позволяет обнулять ошибки инерциальной навигации и сокращать возможные варианты местоположения в алгоритме связывания данных с картой. Сам алгоритм определения типа движения приведен на рисунке

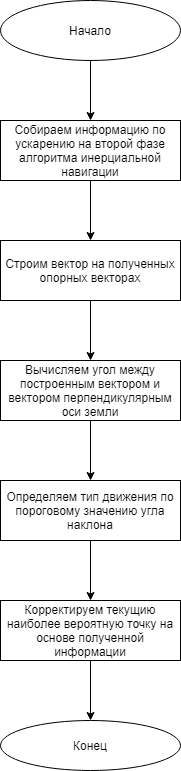


Рисунок Алгоритм определения типа движения

Архитектура взаимодействия модулей:

Для поддержания работы всего многообразия алгоритмов в рамках одного приложения была разработана гибкая многопоточная система взаимодействие модулей инерциальной навигации, связывание данных с картой, и навигации, основанной на машинном зрении.

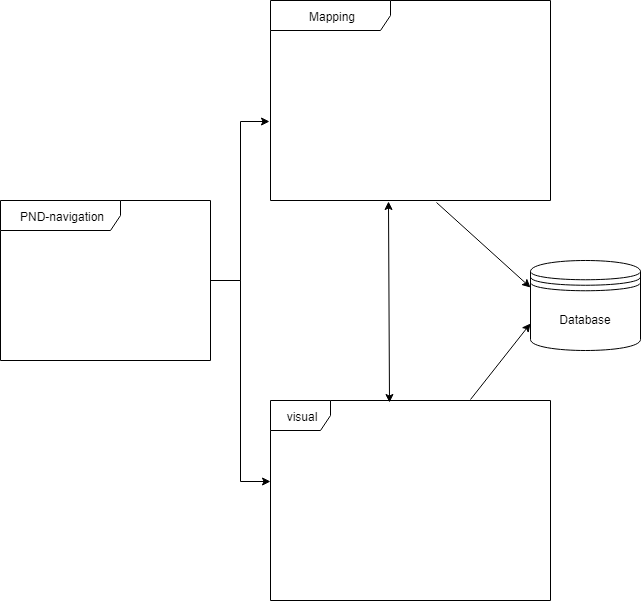


Рисунок 3 общая конфигурация модулей

Идея в том, что основным источником информации является именно инерциальная навигация, остальные алгоритмы вспомогательными и призваны минимизировать ошибку, полученную на первом этапе. Данные полученные на этапе распознавания шага отправляются в систему связывания их с картой, которая возвращает возможные точки, где может находится пользователь, потом данные отправляются так же в визуальную навигацию и в ответ получаем так же набор точек, пересекая результирующие множества, получаем наиболее вероятное место положения пользователя.

Стоит так же заметить, что система расширяема, то есть можно добавлять новые алгоритмы с новыми представлениями данных доступных в среде обработке. Так, например, был добавлен алгоритм отслеживание движения пользователя на основе WIFIсигнала (глава 5).

Реализация метода инерциальной навигации:

В качестве основного метода инерциальной навигации был выбран метод трех фазового контроля, с поддержкой шаблона движения. Метод для пересчета движения работает асинхронно относительно основного потока.

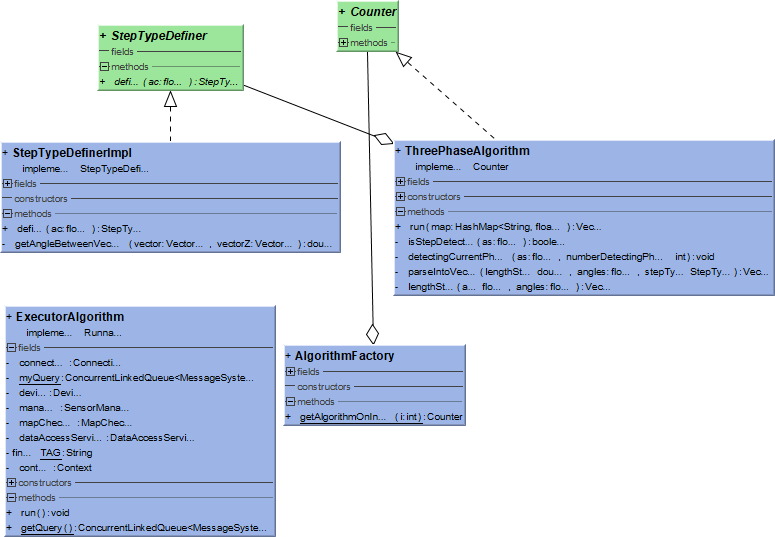


Рисунок 4 диаграмма классов инерциальной навигации

ExecutorAlgorithm – агрегирует в себе необходимые действия которые подложна совершить инерциальная навигация прежде чем передать работу алгоритму корреляции ошибки. Нужный алгоритм определения длины шага выбирается с помощью AlgorithmFactory. Так же в процессе определения шага, алгоритм так же предоставляет информацию о типе шага, реализация этого подпроцесса инкапсулирована в StepTypeDefiner. Основной алгоритм инерциальной навигации (3-фазный) инкапсулирован в класс ThreePhaseAlgorithm

Реализация метода корректировки ошибок:

После получения информации от инерциальной навигации ExecutorAlgorithm передает управление алгоритму корректировки ошибок инерциальной навигации (описание алгоритма приведено в главе 3).



Рисунок 5 Диаграмма классов алгоритма контроля движения с использованием поэтажного плана

Взаимодействие со всеми частями этого модуля производится через ProbabilityMapChecker – этот класс содержит в себе не посредственно калькулятор, который пересчитывает вероятности выбранных точек в процессе работы алгоритма, а также несколько вспомогательных классов, которые содержат в себе различные геометрические операции для вычисления углов и положения отрезков на карте. А именно, CalculatorProbability – вычисляет вероятность конкретной точки имея информацию об расстоянии точки до ближайшего отрезка пути и угол между вектором пути движения и вектором, полученным в ходе работы инерциальной навигации. LineSegmentCustom – инкапсулирует геометрические операции над векторами, необходимые в ходе вычисления вероятностей системы точек. CurrentStateWay – хранит состояние текущего набора системы и результаты вычисления алгоритма в рамках итерации.

1. Обеспечения надежного распознавания объектов из видео потока для коррекции местоположения пользователя

Основная проблема инерциальной навигации, что пользователь должен сам определять начала отсчета или стартовую точку местоположения. Для подобных целей был разработан алгоритм распознавания местоположения пользователя за чёт визуальной навигации. То есть посредством видеокамеры алгоритм распознает контрольные элементы изображения. В приведенной работе это различные таблички ориентации пользователя (рисунок 2).



Рисунок 6Пример таблички ориентации

Общая схема работы алгоритмов визуальной навигации выглядит следующим образом:

1. Выделить последовательность стоп-кадров (VideoFlowAccess).
2. Подготовка изображения (Image Processing).
3. На основе известных точек расположения шаблонов определить местоположение пользователя (DefineLocation).

Детальная архитектура распознавания:

Ниже представлена схема взаимодействия блоков алгоритма распознавания место положения пользователя, на основе графической информации, или визуальная навигация. Можно выделить 3 основные блока взаимодействия, которые упомянутые в начале главы (Рисунок 3).

* VideoFlowAccess
* Image Processing
* DefineLocation

Взаимодействие между блоками выстроено конвейерном формате. Выход каждой части является входом последующей.

## Выделить последовательность стоп-кадров

Эта часть алгоритма отвечает за выделение кадра из видео потока, и вызова следующей части. Источник видео информации выбирветься за счёт Фабрики (Рисунок 4). После из полученого источника выбираеться кадр в настраивывемый промежуток времени. Адаптер предоставляет способ доступа к видеокамере.

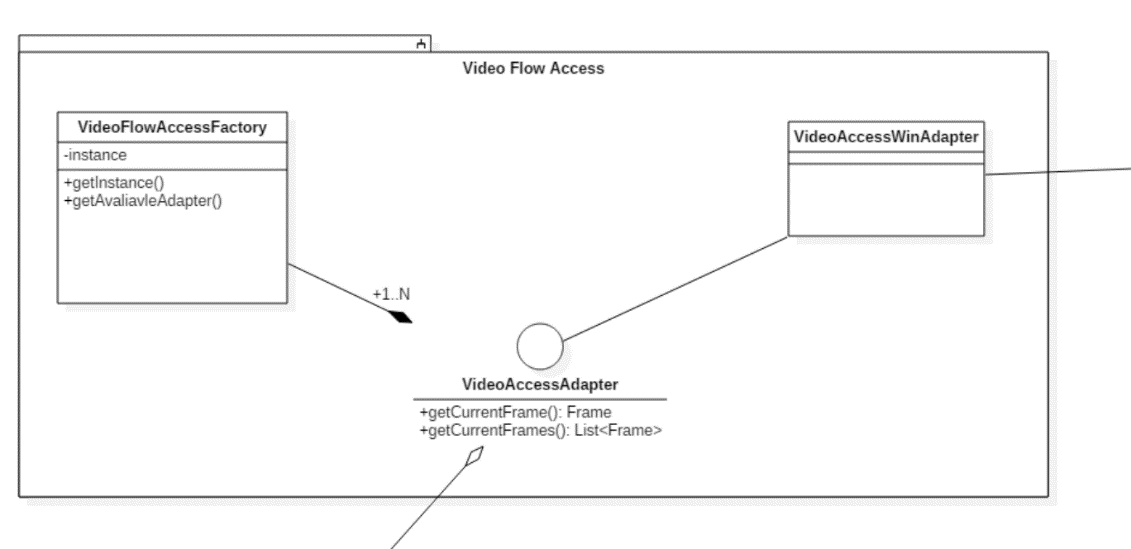


Рисунок 7 VideoAccess

## Подготовка изображения

Далее выбранное изображение фильтруется, (обесцвечивается). Обработка изображения происходит с помощью конвейера, то есть изображение проходит несколько стадий обработки, после чего может быть использована в рамках алгоритма распознавания образов. Все необходимые алгоритмы, которые нужно подвергнуть выбранное изображение. В рамках процесса обработки к изображению применяется алгоритм обесцвечивания и выделение границ.

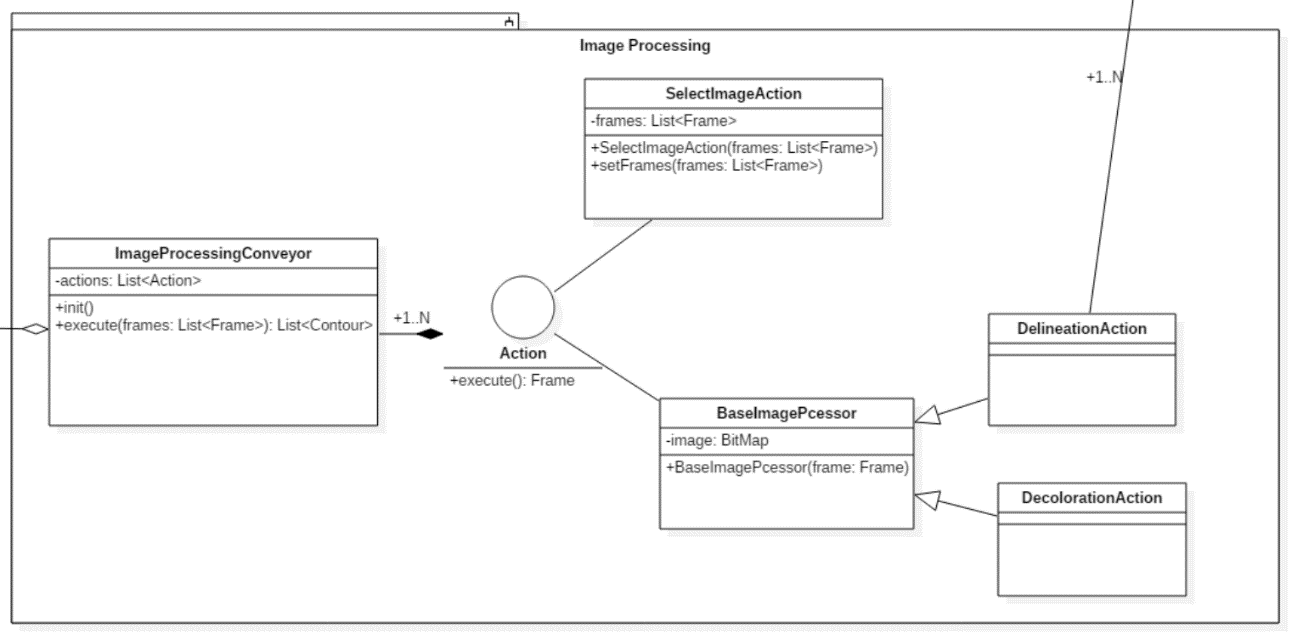


Рисунок 8 Image Processing

## На основе известных точек расположения шаблонов определить местоположение пользователя

После из полученного изображения строиться штриховая, а затем контурная модель, и выбираются те шаблоны, которые наиболее близки к полученным (Рисунок 6).

Зная номера шаблонов, система может выделить те области в который может находиться пользователь. Алгоритм также использует конвейерный подход: раскладываем полученное изображение на шаблонны, затем зная эталонные изображения получаем изображения получаем те шаблоны, которые были найдены на подданном изображении, после чего по найденным шаблоном модно определить возможные точки движения, делается это через базу данных.

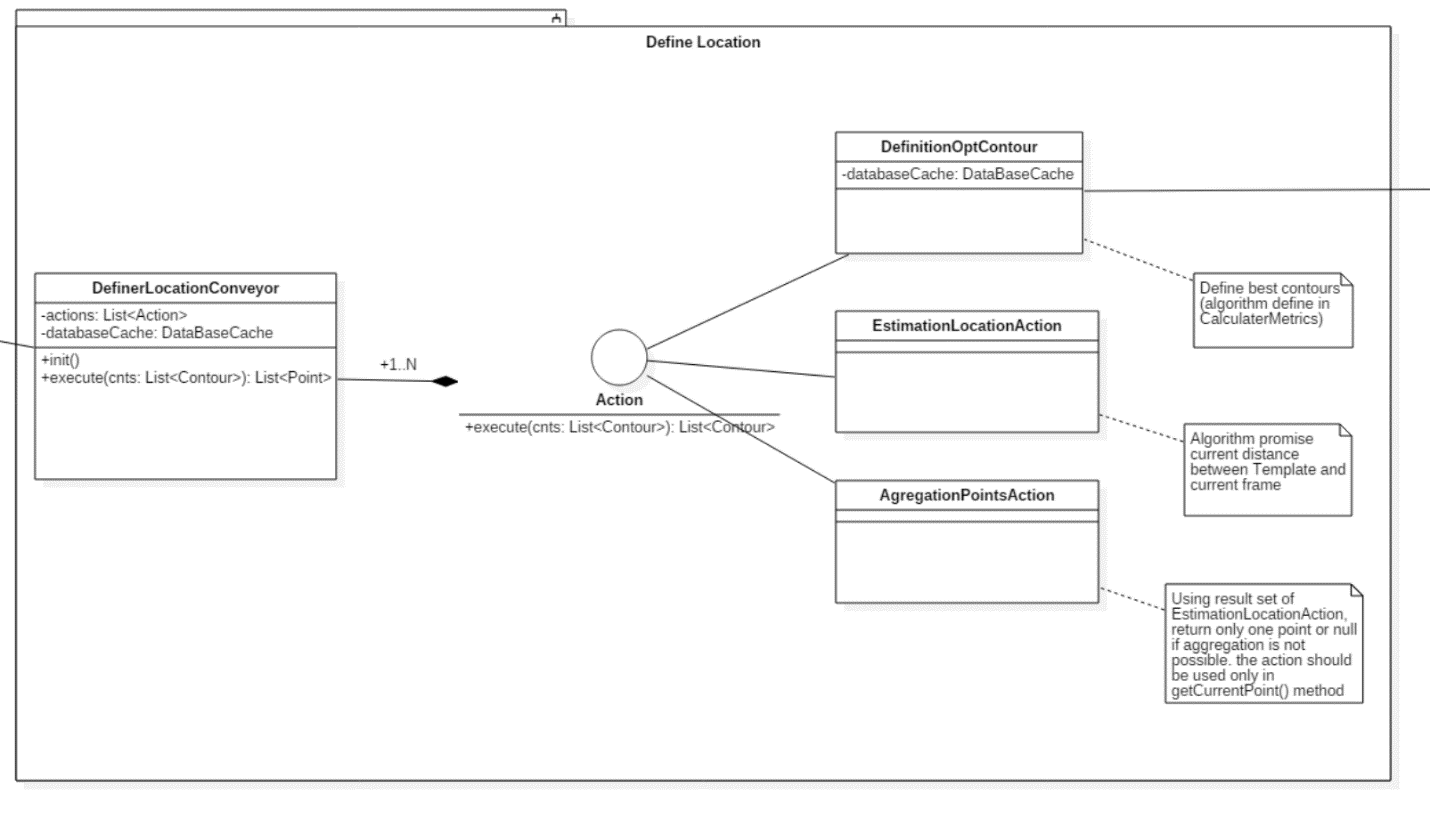


Рисунок 9 Define Location

1. Адаптация для мобильных платформ и последующая оптимизация алгоритмов распознавания объектов из видеопотока

В рамках рассматриваемой задачи было принято решение переноса программных продуктов на мобильную платформу.

1. Перенос инерциальной навигации
2. Перенос алгоритма контроля движения с использованием поэтажного плана
3. Обеспечение связи с датчиками движения
4. Отображение результатов
5. Перенос захвата видео потока на мобильные устройства
6. Реализация диалоговой системы для определения местоположения

В качестве оптимизации выбранных работы приложения, было выбрана отсечение некачественных изображений до этапа построение штриховой и векторной модели для алгоритмов визуальной навигации.

Архитектура

Существует четыре модуля, которые работают асинхронно (Рисунок 5):

Модуль данных – модуль ответственный за считывания данных с датчиков и отправка их модулю расчётов.

Модуль вычислений – модуль ответственный за вычисления по полученным данным (вычисление длины шага).

Модуль коррекции – модуль корректирует результаты вычисления модуля, с помощью карты

Модуль отображения – модуль отрисовывает полученные данные на экран.

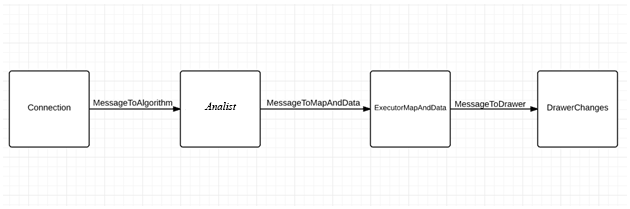


Рисунок 10 Архитектура приложения

В основе модуля коррекции[3] лежит алгоритм, представленный в пункте связывания полученных данных с картой.

Основное внимание стоит уделить модулю вычислений, поскольку, в этом модуле реализуются основные алгоритмы вычисления длины шага

### Архитектура модуля вычислений

Архитектура модуля вычисление шага (Рисунок 6), представлен ввиде паттерна “Пул потоков”, где всю аналитическую работу выполняет класс Analist. Архитектура работы модуля приведена ниже.

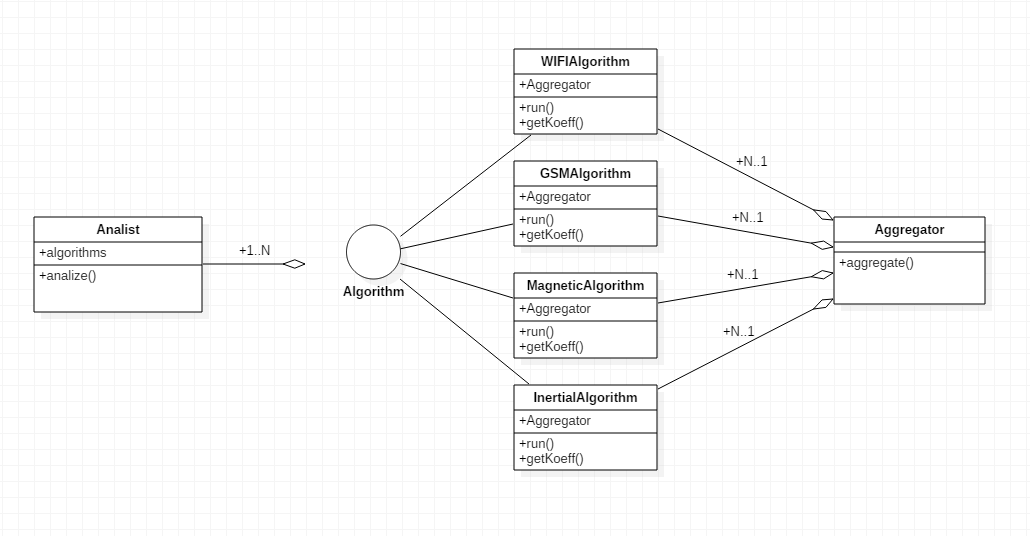


Рисунок 11Архитектура вычислительного модуля

#### Анализ

Analist- ответственный за формирования массива алгоритмов, которые будут учитываться при определении текущего места положения. Каждый алгоритм обладает информацией о том, доступен ли в текущей конфигурации карты. И анализ состоит в том что, перед запуском вычисления, Analist[2.5] проверяет доступность конкретного алгоритма, после чего, запускает его.

#### Архитектура реализации связи с картой

Реализация алгоритма на основе знания о карте.Путь движения считывается в формате графа, где вершинами являются точки изменения траектории движения (повороты), а ребра – линейные части траектории движения. Любой произведенный шаг связывается с картой, то есть вектор, полученный с помощью вычислений (длина + ориентация) отображается на карту, путем вычисления угла между единичным вектором на карте и текущим вектором, полученным путём вычислений. После чего результирующий вектор умножается на матрицу поворота, где угол равен альфа.

Далее, вычисленная точка сопоставляется с каждым отрезком, который связан с рассматриваемым отрезком, с расчётом на то, что поворот пользователя не был распознан. На каждом таком перекрестке создается массив текущих интервалов, по которым рекурсивно применяется приведенный алгоритм, так же в случае перекрестка создается стек движения по каждому направлению и точка разделения, это информации необходима для восстановления движения в случае ошибки (Рисунок 7).

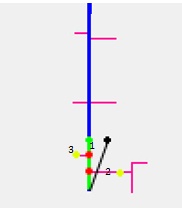


Рисунок 12Второй шаг алгоритма

В дальнейшем движении возможны два случая: первый, что пользователь продолжает идти по выбранному маршруту. Тогда точки, полученные на основе дополнительных путей (отмечены желтым цветом) не попадают в отрезки карты, означает что пользователь не выбрал этот путь и не совершал поворот (Рисунок 8).

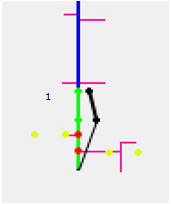


Рисунок 13Третий шаг алгоритма

Второй случай, если поворот был распознан, и пользователь не может продолжать движение по выбранному маршруту. Товероятность нахождения пользователя на отрезках 1 и 3 уменьшается, что приводи к обновлению пути и пересчёту оптимального маршрута (Рисунок 9).

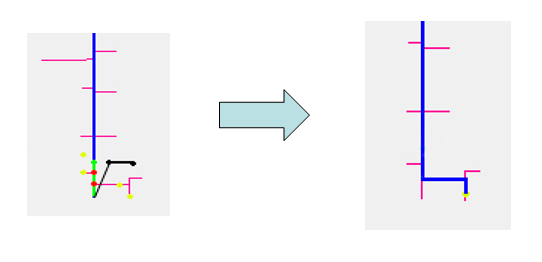


Рисунок 14Обновление пути движения

Описание алгоритма выбора ключевого кадра

В силу длительный вычислений штриховой и векторной модели изображения, и более длительного алгоритма сравнения каждого шаблона с векторным представлением изображения, было принято решение не отдавать все кадры с видео потока на декомпозицию шаблонов, а выбирать наиболее удачные кадры перед запуском процесса распознавания.

Основным критерием отсечения изображения был выбран размытия изображения, что позволило использовать алгоритмы фильтрации изображения.

В литературе популярен простой подход[8] определения степени размытости изображения, который, однако, даёт вполне приемлемые результаты. В этом случае степень размытия оценивается по каждой горизонтальной и вертикальной строке изображения. Обычно бывает достаточно только одной оценки: либо по горизонтали, либо по вертикали. Таким образом, основная часть задачи оценки размытия сводится к одномерной задаче оценки размытия внутри строки. Иными словами, оценивается размытие некоторых одномерных графиков яркости, что существенно проще двумерной задачи оценки размытия. Заключение о размытии всего изображения делается на основе анализа оценок размытия по строкам и столбцам. В простейшем случае численная оценка размытия двумерного изображения даётся как минимум численных оценок размытия по всем строкам и столбцам. Для простоты, дальнейшие рассуждения будут производиться только для строк изображения, так как необходим быстрый способ выявить размытость(Рисунок 10). Стоит заметить, что анализируется не весь кадр, а только верхняя половина кадра.

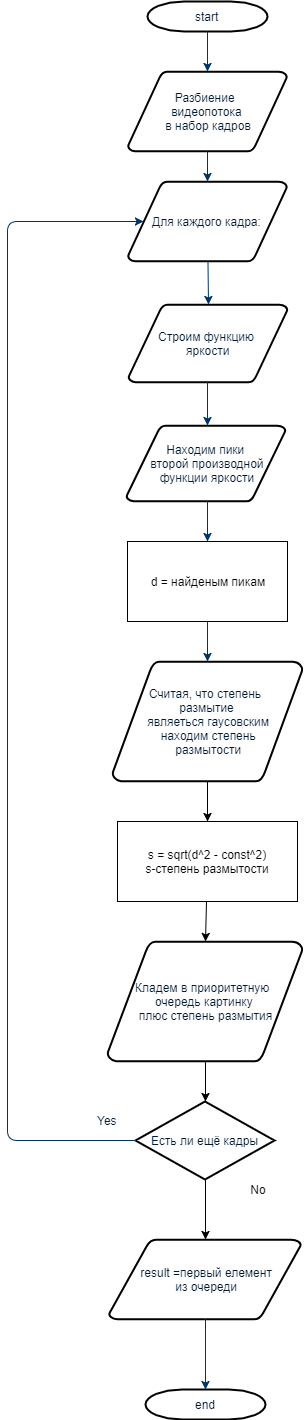


Рисунок 15Блок схема алгоритма

### Архитектура:

Ниже представлена схема взаимодействия блоков алгоритма распознавания место положения пользователя, на основе графической информации, или визуальная навигация. Можно выделить 3 основные блока взаимодействия, которые упомянутые в начале главы.

* VideoFlowAccess
* Image Processing
* DefineLocation

Взаимодействие между блоками выстроено конвейерном формате. Выход каждой части является входом последующей. Стоит так же заметить, что реализация непосредственно распознавания образов велась на с++ языке, поэтому для поддержки алгоритмов распознавания, была введена прослойка между Image Processing и Define location частями алгоритма.

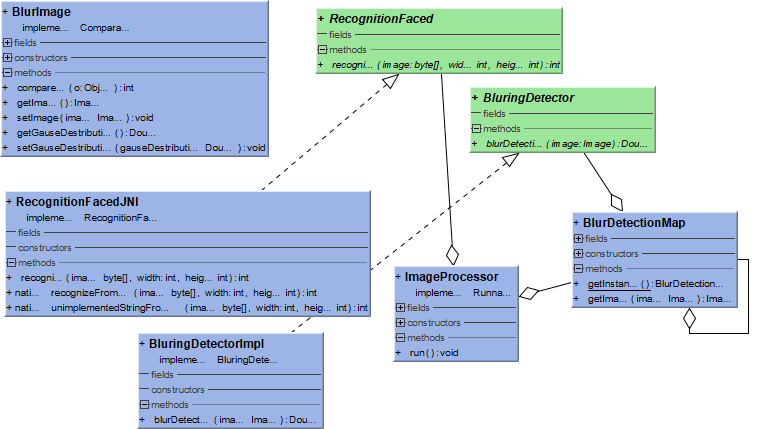


Рисунок 16 Image Processing

На вход в процесс image processing приходит набор изображений, полученных с видео потока, для каждого изображения строится (BluringDetector) гаусовское представления размытия(BlureImage). После чего изображение заносится приоритетную очередь(BlurDetectionMap), время жизни которой ограничено квантом времени для сбора кадров, и после разбора всех полученных изображений выбирается наиболее подходящие для анализа(ImageProcessor), и отправляется в алгоритм разбора на шаблоны и выявление соответствий выбранным шаблонам.

1. Программное обеспечение клиент-серверной системы навигации

Перенос работы с базой данных, которая хранит информацию о поэтажном плане, на выделенный сервер и обеспечение алгоритмов выбора карты движения и оптимального пути движения. При этом для поддержания автономности приложение было принято решении кешировать некоторый объем информации на устройстве клиента.

Архитектура

Ниже будут представлены диаграммы классов для реализации общения клиента и сервера, и забора необходимой информации для работы с поэтажным планом имеющихся алгоритмов.

### Архитектура клиента

Для связи клиента с сервером было разработано следующие основные сущности, которые собирают и основные объекты, связанные с картой и поэтажным планом (Рисунок 11):

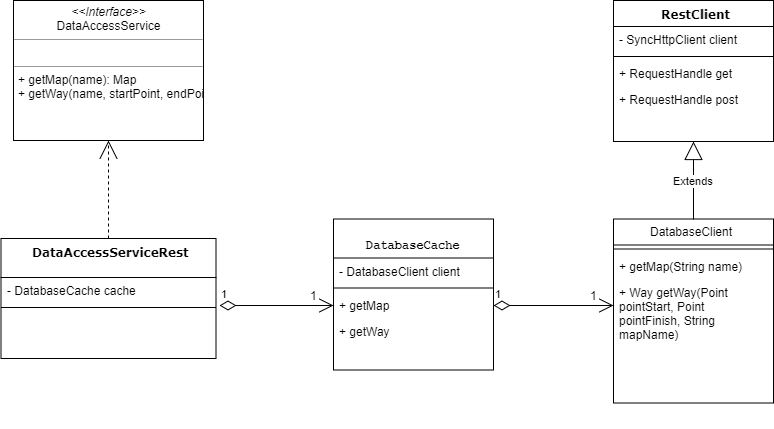


Рисунок 17Диаграмма классов клиента

### Архитектура сервера

Сервер представляет собой два основных блока для организации выполнения задач клиента (Рисунок 12):

1. Блок работы с базой данных
2. Блок расчета оптимального пути

И некий RESTAPIкоторый организует и подготавливает информацию для вышеперечисленных блоков, а так же формирует результат.

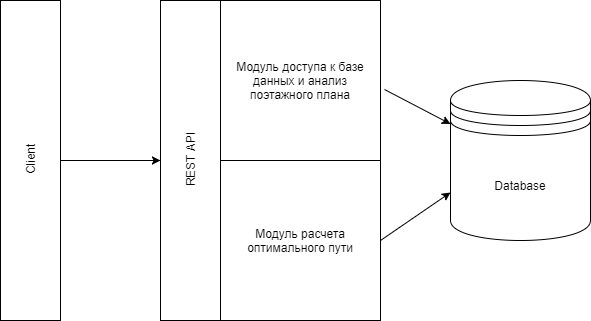


Рисунок 18Архитектура Сервера

Открытые точки доступа на сервере

1. **GET: http://URL/map?name=”plan\_name”**

Возвращает всевозможные пути движения, принадлежащие конкретному поэтажному плану. Имя поэтажного плана задается в параметрах name. Результат запроса Модель: «Карта» (имя карты, линии движения)

1. **POST: http://URL/way?name=”plan\_name”**

Возвращает оптимальный путь движения, принадлежащие конкретному поэтажному плану. Имя поэтажного плана задается в параметрах name. Точки начала и конца движения передаются в теле запроса. Модель для данного RESTAPIесть «Путь»(имя карты, токи старта и финиша, линии движения)

Протокол общения клиента и сервера.

Информация передается в jsonформате. И Сериализуется и десериализуется стандартными средствами платформ.

[

{

"a":{"x":"103,4","y":"389,2"},

"b":{"x":"103,4","y":"378,7"}

},

{

"a":{"x":"103,4","y":"378,7"},

"b":{"x":"103,5","y":"369,7"}

},

{

"a":{"x":"103,5","y":"369,7"},

"b":{"x":"103,6","y":"341,5"}

},

]

1. Литература

[1] J. Kim, H. Jang, D. HwangandC. Park, AStep, StrideandHeadingDetermination for the Pedestrian Navigation System // Journal of Global Positioning Systems, 2004, Vol. 3, No. 1-2, 273-279.

[2] Bogdan Muset, Simina Emerich, Distance Measuring using Accelerometer and Gyroscope Sensors // / Carpathian Journal of Electronic and Computer Engineering 5, 2012, pp. 83-86.

[3] Muhammad Irshan Khan, Design and Development of Indoor Positioning System: For Portable Devices // LAP LAMBERT Academic Publishing, 2013, P. 108.

[4] C.Lukianto and H.Sternberg, “Overview of Current Indoor Navigation Techniques and Implementation Studies” // Bridging the Gap between Cultures, Marrakech. - 2011, 14 p.

[5] Kun-Chan Lan; Wen-Yuah Shih Estimating Step Distance Using Simple Harmonic Motion // 2012 IEEE 75th Vehicular Technology Conference (VTC Spring), 2012, pp. 1 - 5.

[6] Yu.G. Vasin, M.P.Osipov, S.V. Muntyan, E.A. Kustov, “Procedural Modeling and Interactive 3D Visualization of Objects of the Internal Structure of Buildings and Facilities” // Pattern Recognition and Image Analysis, 2015, Vol. 25, No. 2, pp. 278–280.

[7] Yu. G. Vasin, M. P. Osipov, A. Egorov, and Yu. V. Yasakov Autonomous Indoor 3D Navigation // Pattern Recognition and Image Analysis, 2015, Vol. 25, No. 3, pp. 373–377.

[8] IMAGE BLUR ESTIMATION P.P. Koltsov Scientific-Research Institute for System Studies of the RAS

[9] Tips & Tricks: Fast Image Filtering Algorithms Alexey Lukin Department of Computational Mathematics and Cybernetics Moscow State University, Moscow, Russia

[10] DETECTION AND ESTIMATION OF IMAGE BLUR by HARISH NARAYANAN RAMAKRISHNAN

[11] Blur Detection in Video Stream using Filtering Algorithm, Rupali D.Pashte, Rais Mulla, Mahendra Pawar, International Journal of Science, Engineering and Technology Research (IJSETR), Volume 5, Issue 1, January 2016

[12] Дуда Р., Харт П. Распознавание образов и анализ сцен / Пер. с англ. – М.: Мир,1976.512 с. 2.

[13] Лебедев Л.И. Теоретические основы корреляционно-экстремальных контурных методов распознавания // Математические методы распознавания образов (ММРО-15). – 15-ая Всеросс. конф.: Сборник докладов. – М.: Изд-во ООО «МАКС Пресс», 2011. С. 338–341.