



ИПМ им.М.В.Келдыша РАН • [Электронная библиотека](#)

[Материалы защиты](#) • [Сведения о диссертации](#)



**[Пестун М.В.](#)**

Методы построения  
навигационных описаний  
маршрутов для  
картографических  
компьютерных систем

*Диссертация*

**Рекомендуемая форма библиографической ссылки:** Пестун М.В. Методы построения навигационных описаний маршрутов для картографических компьютерных систем: дис. ... канд. физ.-мат. наук: 05.13.11. М., 2015. 115 с. URL: <http://library.keldysh.ru/diss.asp?id=2015-pestun>

Институт Прикладной Математики имени М.В. Келдыша  
Российской Академии Наук

На правах рукописи

Пестун Максим Вадимович

**Методы построения навигационных  
описаний маршрутов для картографических  
компьютерных систем**

05.13.11 – Математическое и программное обеспечение вычислительных машин,  
комплексов и компьютерных сетей

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель:  
д. ф.-м. н., проф. Галактионов Владимир Александрович

Москва – 2015

# Оглавление

Введение.....	4
Глава 1. Современные методы построения и распознавания описания маршрута	12
1.1 Введение в предметную область .....	12
1.1.1 Электронная картография и навигация .....	12
1.1.2 Подзадачи построения и распознавания описания маршрута .....	14
1.1.3 Когнитивная навигация.....	16
1.1.4 Промежуточные выводы.....	17
1.2 Анализ существующих навигационных устройств и систем.....	17
1.2.1 Автомобильные навигационные устройства .....	18
1.2.2 Мобильные навигационные устройства.....	21
1.2.3 Носимые навигационные устройства .....	25
1.2.4 Стационарные навигационные устройства .....	28
1.2.5 Онлайн-навигационные системы.....	30
1.3 Основные методы построения и распознавания описания маршрута.....	33
1.3.1 Пошаговая инструкция.....	34
1.3.2 Изображение траектории .....	35
1.3.3 Использование опорных точек.....	36
1.4 Выводы к первой главе и выбор направления исследования.....	38
Глава 2. Предложенные методы .....	40
2.1 Построение текстового описания маршрута.....	40
2.1.1 Описание алгоритма .....	41
2.1.2 Хранилище данных, словарей и объектов.....	54
2.1.3 Вычисление расстояний между точками на планете .....	57

2.1.4 Проблема использования знакомых пользователю мест и путей.....	58
2.1.5 Проблема склонений слов.....	60
2.1.6 Проблема придания «человечности» формируемому тексту.....	64
2.2 Распознавание текстового описания маршрута.....	66
2.2.1 Описание алгоритма.....	66
2.2.2 Проблема неточности введенных пользователем расстояний.....	68
2.2.3 Проблема нескольких подходящих под описание маршрутов.....	69
2.3 Выводы ко второй главе.....	69
Глава 3. Программная реализация.....	71
3.1 Требования к программной реализации.....	71
3.2 Архитектура программного решения.....	77
3.3 Критерии выбора веб-фреймворков.....	79
3.4 Проблема повторного использования функционального кода.....	80
3.5 Проблема адаптивного интерфейса пользователя.....	81
3.6 Проблема сохранения пользовательских данных.....	82
3.7 Функциональные возможности.....	84
3.8 Практическое применение.....	89
3.9 Сравнение.....	95
3.9.1 Построение описания маршрута.....	97
3.9.2 Распознавание описания маршрута.....	100
3.9.3 Использование системы навигации в офисе SkyLight.....	103
3.10 Выводы к третьей главе.....	104
Заключение.....	105
Литература.....	106
Список рисунков.....	114

# Введение

**Актуальность работы.** В современном мире интеграция вычислительной и роботизированной техники в повседневную жизнь становится заметнее с каждым днем. Многие механические и аналоговые приборы заменяются электронными, приобретают средства для взаимодействия с другими устройствами, объединяются в сети и образуют так называемый интернет вещей. В этой новой среде человеку отводится управляющая роль, он определяет задачи, которые далее выполняются роботами, компьютерами и другими электронными устройствами. Таким образом, в сферу общения человека, традиционно состоящую из окружающих его людей, включаются и роботизированные системы. Поддержание с ними диалога осложняет специфика самих устройств, их стремительное развитие и необходимость постоянного переобучения. В связи с этим возникает научная проблема на прикладном уровне по разработке интерфейса для взаимодействия человека и роботизированных систем подобного рода.

Важной частной проблемой является область передачи знания о маршруте. Знание о маршруте может передаваться как от человека к роботизированной системе, так и в обратном направлении – от роботизированной системы к человеку. Для передачи таких знаний требуется специальный формат описания маршрута, удобный для понимания человеком и распознаваемый роботом. Помимо этого требуется возможность персонализации передаваемых данных под конкретного пользователя. Ее наличие позволит сократить описание, упростить его понимание и увеличить вероятность правильного распознавания при вводе.

На сегодняшний день существует много различных систем, частично решающих данную проблему – это автомобильные навигаторы, мобильные и стационарные платформы, веб-приложения и другие. Далее по тексту они называются общим именем – компьютерными навигационно-картографическими системами.

Такие системы предлагают своим пользователям возможности по нахождению оптимального пути из точки А в точку В и выводят, преимущественно на экран, инструкции по перемещению, выполнение которых проведет пользователя по маршруту. Эти инструкции обычно не содержат знакомых пользователю ориентиров. Ввод конкретного маршрута в такие системы либо невозможен (обычно предлагается использовать полностью автоматическое построение), либо сильно ограничен.

Можно выделить следующие *недостатки существующих навигационно-картографических систем*:

- карты не подстраиваются под пользователя и не выделяют важную только для него информацию, усложняя тем самым их использование;
- навигационно-картографические системы мало приспособлены для слабовидящих или незрячих пользователей;
- пошаговое описание маршрута обладает слабой выразительностью и не имеет персонализации, что осложняет восприятие;
- методы ввода конкретного маршрута в компьютер малофункциональны или отсутствуют, что не дает возможности точно задать конкретный путь;
- методы хранения навигационных карт обладают избыточной информацией; с целью уменьшения объема необходимо сконцентрироваться только на важных для пользователя деталях, расставив им приоритеты.

На основе анализа перечисленных выше конкретных недостатков и общей проблемы передачи знания о маршруте *ставится научная задача* по разработке методов построения персонализированных навигационных текстовых описаний маршрутов для картографических систем, позволяющих упростить

взаимодействие человека с компьютерными и роботизированными платформами на уровне обмена знаниями.

Построение и ввод описания маршрута в удобном (и, по возможности, персонализированном) для человека виде актуальны в следующих задачах:

- взаимодействие с роботизированными системами;
  - На текущий момент идет стремительное развитие антропоморфных роботов, лишенных графических экранов и умеющих взаимодействовать с пользователями только при помощи голоса. Вопрос описания маршрута в словесной форме самим роботом и понимания того, какой путь имел в виду человек, становится сейчас актуальным.
- указание компьютеру о следовании по специальному маршруту;
  - На текущий момент навигационные системы для широкой аудитории не предлагают возможности легко и удобно задать конкретный маршрут, основанный на персональных знаниях пользователя, они могут лишь сами предлагать варианты для выбора. Однако порой необходимо узнать время в пути именно по конкретному маршруту или сообщить автоматическому автомобилю (полностью управляемому роботом) о желаемом пути проезда. Удобным решением будет возможность описать маршрут в приблизительном виде и попросить компьютер предложить наиболее подходящие под описание варианты, при этом пользователь может оперировать знакомыми только ему объектами, такими как «дом», «работа» и т.д. Существующие навигационные системы для решения этой задачи используют интерфейс, в котором пользователь может поставить на карте транзитные точки, через которые

должен проходить автоматически проложенный маршрут. Однако такой ввод данных становится слабо актуальным в связи с развитием голосового ввода информации («ОК Google», «Siri»), что подталкивает к разработке универсального решения.

- навигация в городском окружении;
  - Использование навигационных систем (например, автомобильного навигатора) частично решает проблему поиска нужного места в сложной системе дорог, однако их рекомендации не всегда бывают удобными для водителя, из-за чего ему требуется отвлекаться от дороги на монитор устройства с изображением карты и нарисованной поверх нее траектории движения. Инструкция «поверните налево через 312 метров» может быть оформлена в более удобной форме «поверните налево за площадью».
- навигация внутри зданий со сложной планировкой;
  - В современном мире здания стали иметь очень большие размеры, постоянно расширяются и достраиваются. В связи с этим часто бывает осложнена навигация по ним. Система таблиц и указателей упрощает ее, но не всегда бывает достаточной: указатели на все интересующие человека места физически невозможно разместить повсеместно. На сегодняшний день наибольшие трудности вызывают следующие здания: торговые центры, аэропорты, многоэтажные парковки на тысячи автомобилей и другие крупные сооружения. Иногда встречаются здания с планировкой, при которой, например, на второй этаж можно попасть только с третьего, но не с первого этажа. Решением, упрощающим навигацию, является расстановка небольших

терминалов, на которых пользователь может выбрать интересующий его объект (конкретный магазин, парковочное место, кабинет), и система расскажет ему, как до него добраться. Частично описание может быть текстовым, оформленным в виде прямой речи человека, объясняющего «как пройти». Такой текст должен быть легко запоминаемым и однозначно понятным.

**Цель диссертационной работы.** Цель диссертационной работы состоит в разработке алгоритмов построения персонализированных навигационных описаний маршрутов для упрощения взаимодействия человека с компьютером в области передачи информации о пути и создании программных средств на их основе для использования в существующих картографических системах.

Для достижения поставленной цели необходимо последовательно решить следующие **задачи**:

- исследовать и провести анализ существующих навигационно-картографических решений;
- исследовать существующие методы построения описания (вывода) маршрута; разработать алгоритм построения персонализированного текстового описания маршрута в удобной для человека форме;
- исследовать существующие методы задания (ввода) маршрута в компьютер; разработать алгоритм преобразования персонализированного текстового описания маршрута в удобное для обработки компьютером представление;
- разработать компьютерную систему, спроектировав ее архитектуру, реализующую данные алгоритмы.

**Научная новизна.** Получены следующие новые результаты в области передачи навигационной информации:

1. Разработаны и реализованы новые алгоритмы построения и распознавания персонализированного когнитивного текстового описания маршрута, упрощающие использование навигационно-картографических систем. Предложенные алгоритмы интегрируемы в широкий класс навигационных систем, работающих как на стационарных, так и на мобильных платформах.
  - а. Алгоритм построения текстового описания учитывает персональные знания пользователя об окружающих объектах, его передвижения в прошлом, часто посещаемые места и другую информацию, позволяющую сократить и упростить описание.
  - б. Алгоритм распознавания текстового описания подбирает подходящие под введенный текст маршруты с учетом данных, накопленных о пользователе ранее, что дает возможность значительно уменьшить число ошибочных вариантов. В случае если информации недостаточно для точного определения маршрута, то пользователь может его уточнить дополнительным вводом.
2. Для построения текстовых описаний разработан вспомогательный алгоритм синтеза предложений, основанный на вероятностных величинах, что придает вид живой речи формируемому тексту.
3. На основе проведенных исследований и экспериментов по формированию когнитивной карты в сознании человека и восприятию текстового описания пути предложены новые алгоритмы для хранения и отображения данных о маршруте, учитывающие психологию восприятия картографической информации. Благодаря такому подходу возможно упрощение восприятия пользователем информации о пути, а также уменьшение объема необходимой

информации для осуществления навигации, хранящейся на устройстве.

**Практическая значимость.** Разработанный на основе предложенных алгоритмов программный комплекс используется:

- в летающей автономной роботизированной системе (квадрокоптер), способной огибать препятствия на пути согласно заданной инструкции, описывающей маршрут передвижения в близком к естественному языку виде (использовалась заранее подготовленная карта местности с размеченными объектами). Робот был разработан, собран и запрограммирован командой в составе с автором диссертации для конкурса летающих роботов от компании КРОК.
- в автономном роботе RB-2, предназначенном для использования в комнатных условиях с целью выполнения бытовых задач (принести, проверить, убрать и т.д.), разрабатываемого группой ученых из ИПМ им. М. В. Келдыша РАН.
- в созданной автором навигационной системе Интранета высотного офисного здания SkyLight (Ленинградский проспект, д. 39). Любой сотрудник, работающий в данном здании, имеет возможность получить подробную инструкцию о том, как добраться до рабочего места интересующего его человека, просмотрев информацию о нем в персональном профиле на корпоративном портале; обычные методы навигации (указатели, надписи) работали плохо ввиду сложного зеркального по четырем направлениям расположения рабочих мест (используется методика рассадки open-space).
- картографической системе Карты Mail.Ru для предоставления описания маршрута пользователю в удобном персонализированном виде.

**Апробация работы.** Основные результаты диссертации докладывались на следующих конференциях и семинарах:

1. Международная Конференции по Компьютерной Графике и Зрению GraphiCon'2012 (Москва, факультет ВМК МГУ).
2. Международная Конференции по Компьютерной Графике и Зрению GraphiCon'2014 (Ростов-на-Дону, ЮФУ).
3. Международная Конференции по Компьютерной Графике и Зрению GraphiCon'2015 (Протвино, ИФТИ).
4. Научно-практический семинар «Новые информационные технологии в автоматизированных системах» 2014 года (Москва, НИУ ВШЭ).
5. Научно-практический семинар «Новые информационные технологии в автоматизированных системах» 2015 года (Москва, НИУ ВШЭ).
6. Семинар направления “Программирование” им. М. Р. Шура-Бура в ИПМ им. М. В. Келдыша РАН.

**Публикации.** По результатам работы имеются 9 публикаций [1-9], из них 2 опубликованы в рецензируемых журналах Перечня ВАК [6, 8], 1 публикация входит в библиографические базы Web Of Science и Scopus [1].

**Личный вклад автора.** Содержание диссертации отражает персональный вклад автора в опубликованных работах. Подготовка к публикации полученных результатов проводилась совместно с соавторами, причем вклад диссертанта был определяющим. Все представленные в диссертации результаты получены лично автором.

**Структура и объем диссертации.** Диссертация состоит из введения, 3 глав, заключения и списка литературы. Общий объем диссертации составляет 115 страниц, из них — 102 страницы основного текста, включая 35 рисунков. Библиография включает 80 наименований.

# **Глава 1. Современные методы построения и распознавания описания маршрута**

Современные навигационно-картографические системы имеют много различных форм и представлений, начиная с онлайн-сервисов, доступных на любом компьютере, и заканчивая мобильными устройствами, которые всегда можно взять с собой. Данная глава начинается с введения в предметную область, содержит основные определения и термины. Далее рассматриваются основные подзадачи картографических систем, среди которых выделяются построение и распознавание описания маршрута. Приводится сравнительный анализ существующих навигационно-картографических систем.

## **1.1 Введение в предметную область**

### **1.1.1 Электронная картография и навигация**

Тема электронной картографии и навигации очень актуальна и востребована на сегодняшний день [17], этому способствовало стремительное развитие:

- систем глобального позиционирования (GPS [18], ГЛОНАС [19], BeiDou [20]);
- телекоммуникационных технологий передачи данных на любые расстояния;
- информационных банков данных картографической информации (Google, Яндекс и др.).

Продуктом электронной картографии являются навигационно-картографические системы — распределенные программные комплексы с огромным массивом информации, хранимой в базах данных [21]. Устройства, на которых можно ими воспользоваться, очень многочисленны:

- персональные компьютеры и ноутбуки;
- автомобильные навигаторы;
- мобильные телефоны, смартфоны и планшеты;
- смарт-часы, смарт-очки;
- навигационные стационарные терминалы.

Базы данных существующих систем содержат миллионы статических объектов и огромное количество динамических данных по всему миру:

- названия административных и географических объектов, названия дорог, адреса домов, разметка дорожного движения, установленный допустимый скоростной режим;
- офисы организаций и компаний, социальные учреждения, достопримечательности;
- фотографии, описания, комментарии пользователей, оценки экспертов;
- временные события (например, ярмарки);
- остановки общественного транспорта, линии его движения, расписание;
- текущее состояние движения на дорогах (пробки, заторы, ДТП);
- текущее положение отдельных единиц общественного транспорта.

Навигационно-картографические системы решают следующие задачи:

- предоставление актуальных статических и динамических данных;
- локализация положения пользователя;
- работа с маршрутами.

Работа с маршрутами имеет следующие подзадачи:

- автоматический расчет оптимального маршрута между заданными точками его начала и конца;
- построение описания маршрута (передача информации от системы к пользователю);

- распознавание описания маршрута (передача информации от пользователя к системе);
- отслеживание перемещения по заданному маршруту.

Интерес для данной диссертации представляют подзадачи построения и распознавания описания маршрута. Они имеют готовые решения на сегодняшний день, однако степень их удобства для пользователя находится на недостаточном уровне из-за накладываемых ограничений, рассмотренных ниже.

В тексте будет использоваться аббревиатура *POI*, обозначающая любое место интереса (от английского Point Of Interest): здания, организации, парки, памятники искусства, достопримечательности, остановки общественного транспорта и т.д. Термины *маршрут* и *путь* принимаются в качестве синонимов. У любого пути есть точка *начала (стартовая)* и *конца (финишная)*. Маршрут является *ломаной линией*, представляющей собой один из множества возможных вариантов и описывающий как добраться из начальной точки в конечную, используя один или несколько способов перемещения: пешком, на автомобиле, на самолете, на водном судне и т.д. Между двумя точками может быть любое количество возможных путей, в том числе и нулевое. Ломаная линия состоит из *вершин (точек)*, соединенных прямыми отрезками. Вершины ломаной линии, а также точки начала и конца маршрута имеют географические координаты, задаваемые долготой, широтой и альтитудой (высота точки земной поверхности над уровнем океана), где это имеет смысл.

### **1.1.2 Подзадачи построения и распознавания описания маршрута**

Подзадача построения описания маршрута состоит в создании некоторого представления автоматически найденного пути между наперед заданными точками его начала и конца. Данное представление используется человеком и должно быть [10]:

- легко воспринимаемым — с целью сокращения затрачиваемого пользователем времени для его понимания;

- легко запоминаемым — с целью использования пользователем полученных от системы знаний о маршруте без многократного обращения к ней;
- персонализированным, то есть использующим знакомые для конкретного пользователя представления и POI — с целью упрощения восприятия и запоминания;
- наглядным, то есть использующим визуальную информацию об объектах, встречаемых на пути — с целью задействования одновременно нескольких видов памяти пользователя.

Подзадача распознавания описания маршрута состоит в нахождении компьютерного представления пути по введенным пользователем данным, характеризующим его. Данный метод должен быть [10]:

- лаконичным — от пользователя должно требоваться лишь минимальное количество необходимой информации для точного идентифицирования описываемого маршрута;
- персонализированным — пользователь должен иметь возможность использовать знакомые конкретно ему представления и POI;
- не чувствительным к ошибкам — пользователь может ошибаться, например, в расстояниях, система должна быть лояльна этому;

Выше описаны общие требования к решениям подзадач построения и описания маршрута в навигационно-картографических системах, которые используются для определения конкретных требований к соответствующим алгоритмам, рассматриваемым в данной диссертационной работе.

Забегая немного вперед, стоит сразу определить существующие способы представления описания маршрута, доступные на сегодняшний день. Несмотря на то, что подробный анализ методов изложен далее по тексту, на данном этапе необходимо их определить и дать названия. Таким образом, построение описания маршрута между наперед заданными точками старта и финиша сейчас возможно двумя вариантами:

- *изображение траектории* — представляет из себя вид сверху на карту с нарисованной ломаной линией движения;
- *пошаговая инструкция к перемещению* — последовательность действий (поворотов, разворотов и т.д.), выполнение которых сможет привести от старта пути к его финишу.

### 1.1.3 Когнитивная навигация

Существует много разных определений понятия «когнитивная навигация». В данной диссертации введем следующие определения, которые будут подразумеваться по всему тексту:

- *Когнитивная карта* (от лат. *cognitio* — знание, познание) — образ знакомого пространственного окружения. Когнитивные карты создаются и видоизменяются в результате активного взаимодействия субъекта с окружающим миром. При этом могут формироваться когнитивные карты различной степени общности, «масштаба» и организации (например, карта-обозрение или карта-путь в зависимости от полноты представленности пространственных отношений и присутствия выраженной точки отсчета). Это — субъективная картина, имеющая прежде всего пространственные координаты, в которой локализованы отдельные воспринимаемые предметы [80]. Выделяют карту-путь как последовательное представление связей между объектами по определенному маршруту и карту-обозрение как одновременное представление пространственного расположения объектов. Изначально термин был предложен в 1948 в работе американского психолога Э. Толмена «Когнитивные карты у крыс и человека» [22].
- *Навигация* — процесс управления некоторым объектом в определённом пространстве передвижения; подзадачей навигацией является маршрутизация — процесс определения маршрута

следования в некоторой сети окружения (графе пересечений дорог в городе, графе коридоров и лестниц внутри зданий) [11].

- *Когнитивная навигация* — процесс управления некоторым объектом в определенном пространстве передвижения с учетом его персональных знаний об окружении [11].

Человек строит когнитивную карту в своем сознании при перемещении в пространстве, со временем карта дополняется и уточняется [80]. Когнитивная карта имеет прежде всего пространственные координаты [12], в которых локализованы отдельные воспринимаемые предметы, благодаря которым возможна когнитивная навигация.

#### **1.1.4 Промежуточные выводы**

В начале первой главы выделены две конкретные подзадачи, исследованию которых посвящен последующий текст диссертации: построение и распознавание описания маршрута в персонализированном и удобном для пользователя виде. Даны четкие определения используемых понятий с целью уменьшения разночтений.

### **1.2 Анализ существующих навигационных устройств и систем**

В данном параграфе рассматриваются методы построения и распознавания описаний маршрутов в существующих навигационно-картографических системах. В современном мире технологии очень быстро меняются и развиваются, поэтому необходимо уточнить, что анализ проводился на преимущественно последних версиях программного обеспечения, доступных на момент написания. К сожалению, номер сборки не всегда возможно определить, особенно у онлайн-систем с непрерывным выводом обновлений, с целью упрощения описания он опускается. Рассмотрению подлежали как специализированные навигаторы, преимущественно используемые в автомобилях, так и специальные программы, устанавливаемые на мобильные устройства и превращающие их в

аналоги специализированных навигаторов. Также рассмотрению подлежали носимые устройства (смарт-очки, смарт-часы) и стационарные системы, используемые в крупных зданиях или культурных центрах городов, часто посещаемых туристами.

### 1.2.1 Автомобильные навигационные устройства

Автомобильные навигационно-картографические системы, именуемые в основном навигаторами, сейчас очень распространены, имеют различные форм-факторы, однако всегда состоят из:

- графического экрана, на котором отображается карта, текущее местоположение и проложенный маршрут;
- системы ввода нажатием на экран или вынесенными на корпус клавишами;
- динамика, способного воспроизводить в достаточном для восприятия качестве человеческий голос, озвучивающий инструкции к перемещению.

Рассмотрения подлежали следующие устройства:

- Navitel [23]
- Garmin [24]
- Mio [25]

Навигаторы компании **Navitel** (рис. 1) [23] имеют множество различных реализаций, в том числе для разных платформ. При этом все они похожи по функционалу и внешнему виду пользовательского интерфейса. Система реализует оба варианта описания маршрута: изображение траектории и пошаговая инструкция к перемещению. Ввод описания маршрута невозможен, есть только функционал для задания стартовой и финишной точек маршрута, а также уточнения его дополнительными транзитными точками, что по своей сути разбивает маршрут на несколько последовательных путей.

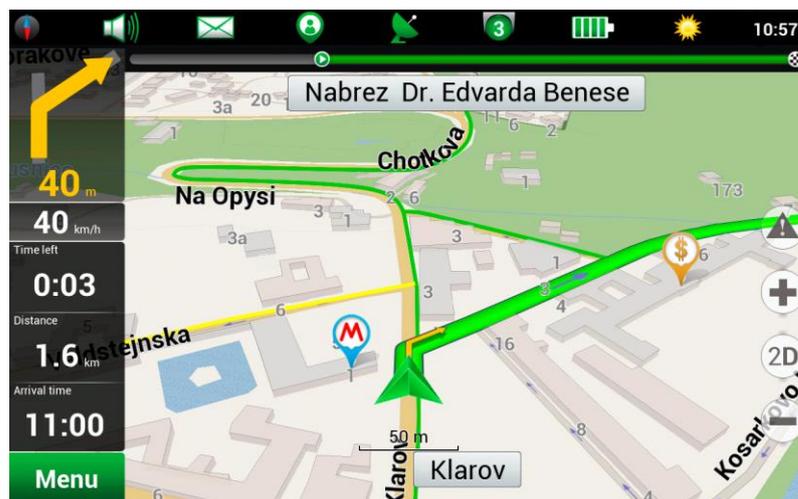


Рис. 1: Пример экрана системы Navitel

Навигаторы компании **Garmin** (рис. 2) [24] также насчитывают большое количество вариантов реализаций и форм-факторов, при этом оставаясь стабильными в плане предлагаемого пользователям функционала и интерфейса пользователя. Система аналогично реализует оба варианта описания маршрута: изображение траектории и пошаговая инструкция к перемещению. Ввод описания маршрута, как и в случае с Navitel, может быть осуществлен только путем задания стартовой и конечной точки с возможностью добавления транзитных мест. Дальнейшее определение траектории пути происходит в автоматическом режиме.



Рис. 2: Пример экрана системы Garmin

Навигаторы компании **Mio** (рис. 3) [25] давно представлены на рынке. В них аналогично существует большое количество вариантов реализаций внешнего вида, но при этом навигаторы остаются постоянными в плане предлагаемых инструментов. Реализованы оба варианта описания маршрута: изображение траектории и пошаговая инструкция к перемещению. Ввод описания маршрута ничем не отличается от других систем и происходит через задание краевых и опорных точек.

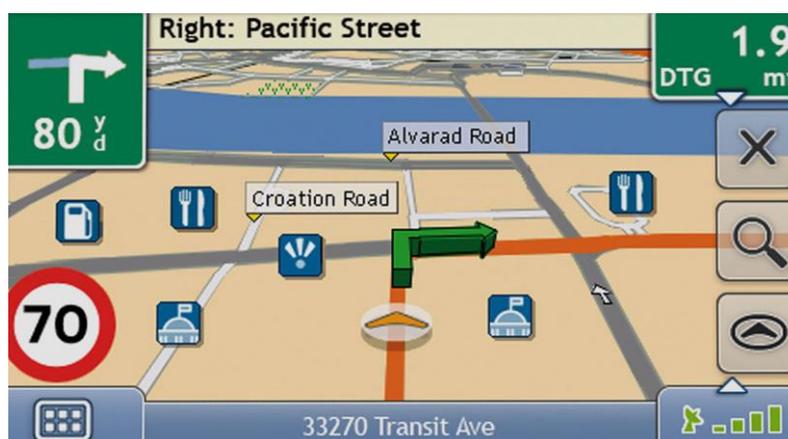


Рис. 3: Пример экрана системы Mio

Общей чертой для большинства (но не для всех) автомобильных навигаторов является работа в автономном режиме, что означает:

- информация о пробках и других динамических событиях недоступна;
- информация по дорогам, знакам и POI обновляется редко и требует подсоединения к компьютеру;
- персональная информация о занесенных в избранное POI и часто используемых маршрутах является локальной и не синхронизируется с другими устройствами пользователя;
- навигаторы автоматически не подстраивают свои алгоритмы под пользователя на основе истории его перемещений и не предлагают средств упрощения описания маршрутов на основе известных ему POI и участков пути.

## 1.2.2 Мобильные навигационные устройства

Смартфоны и планшеты могут решать те же самые задачи, что и автомобильные навигаторы. Для этого требуется установка специального навигационно-картографического программного приложения. Рассмотрению подлежали программы:

- Apple Maps [26]
- Google Maps [27]
- Яндекс Карты [28]
- Яндекс Навигатор [29]

Все рассматриваемые в данном разделе навигационно-картографические системы представляют собой клиентские приложения с хранением информации на удаленных серверах. Таким образом, программная система делится на две части:

- серверная часть:
  - хранит информацию о картах, изображениях, POI и т.д.;
  - прокладывает маршруты;
  - доступна по сети Интернет из любой точки мира;
  - имеет постоянно обновляемую и актуализируемую картографическую информацию в соответствии с изменениями в реальном мире
- клиентская часть:
  - хранит локальные настройки пользователя;
  - кэширует последнюю полученную от сервера информацию на некоторое время, определенное разработчиком, после чего проверяет, есть ли в ней изменения и в случае их наличия запрашивает у сервера обновленные данные;
  - визуализирует полученные от сервера данные в удобном для пользователя виде;

- осуществляет локализацию положения пользователя и отображение его текущего местоположения на карте;
- приложение обновляется реже и требует полную переустановку, производимую операционной системой в автоматическом режиме с согласия пользователя.

Мобильная навигационно-картографическая система **Apple Maps** [26] от одноименной компании Apple работает только на устройствах компании, предлагает уже стандартные способы описания маршрута в виде изображения траектории (рис. 4) и пошаговой инструкции к перемещению (рис. 5), как это показано на иллюстрациях ниже.

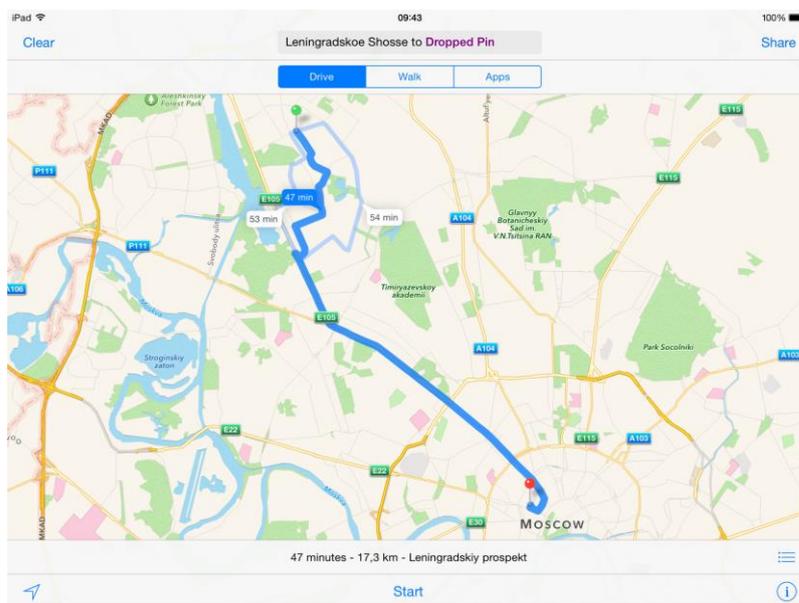


Рис. 4: Пример экрана системы Apple Maps (изображение траектории)

Ввод конкретного пути в системе Apple Maps невозможен, есть только функционал по заданию стартовой и конечной точки маршрута, далее он рассчитывается автоматически на серверной стороне приложения. Добавление промежуточных точек трансфера, как в автомобильных навигаторах, также невозможно. Система Apple Maps не имеет функционала для персонализации при построении описания маршрута.

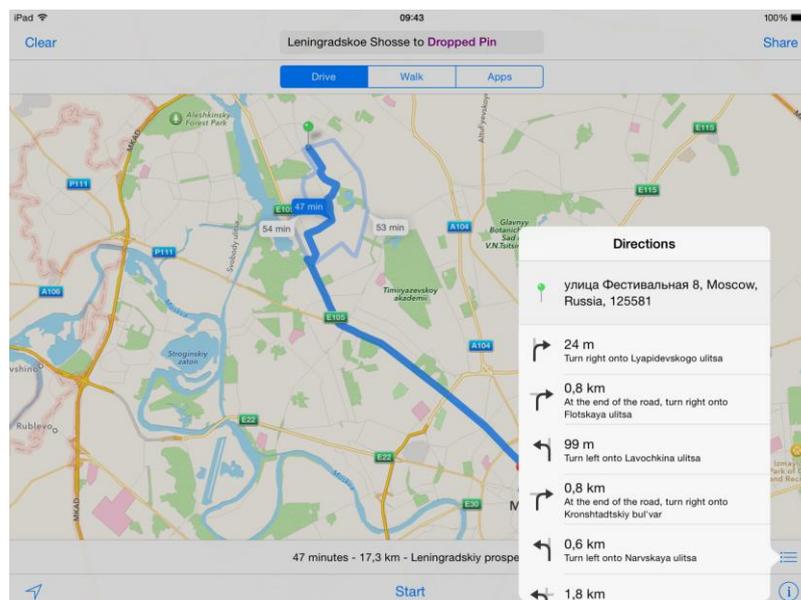


Рис. 5: Пример экрана системы Apple Maps (пошаговая инструкция к перемещению)

Навигационно-картографическая система **Google Maps** [27] от компании Google реализована для множества платформ, в том числе для устройств фирмы Apple и устройств на базе собственной операционной системы Android. Функционал мало отличается от рассмотренной выше системы Apple Maps. Те же самые способы описания маршрута (рис. 6): визуальное представление траектории пути и пошаговая инструкция к перемещению без учета персональных знаний пользователя об окружающих объектах и истории его перемещений в прошлом. Ввод промежуточных точек транзита невозможен.

Система **Яндекс Карты** [28] от компании Яндекс широко распространена в основном на территории России и СНГ. Приложение сконцентрировано на предоставлении справочной информации по расположению объектов и текущему состоянию пробок. Функционал по формированию описания проложенного маршрута возможен только в виде визуального представления траектории (рис. 7). Ввод маршрута возможен только через начальную и конечную точку без функционала по его уточнению.

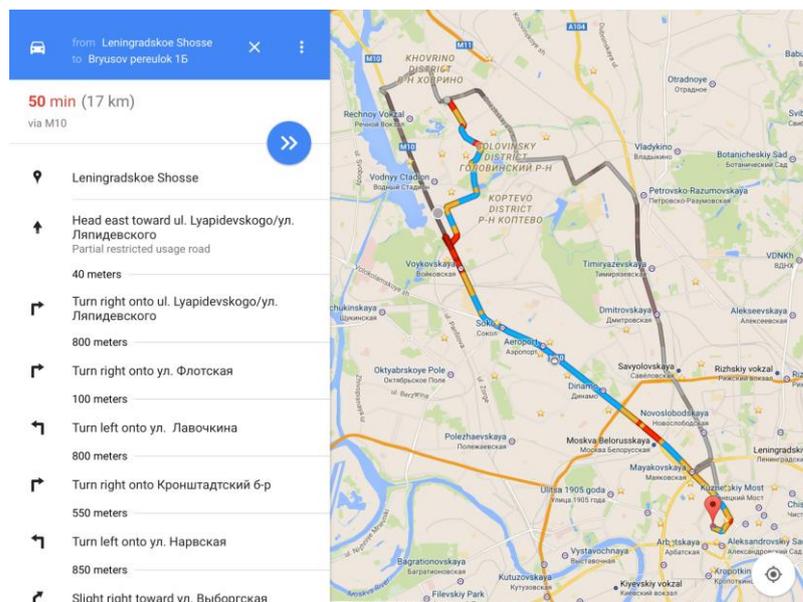


Рис. 6: Пример экрана системы Google Maps (изображение траектории и пошаговая инструкция к перемещению)

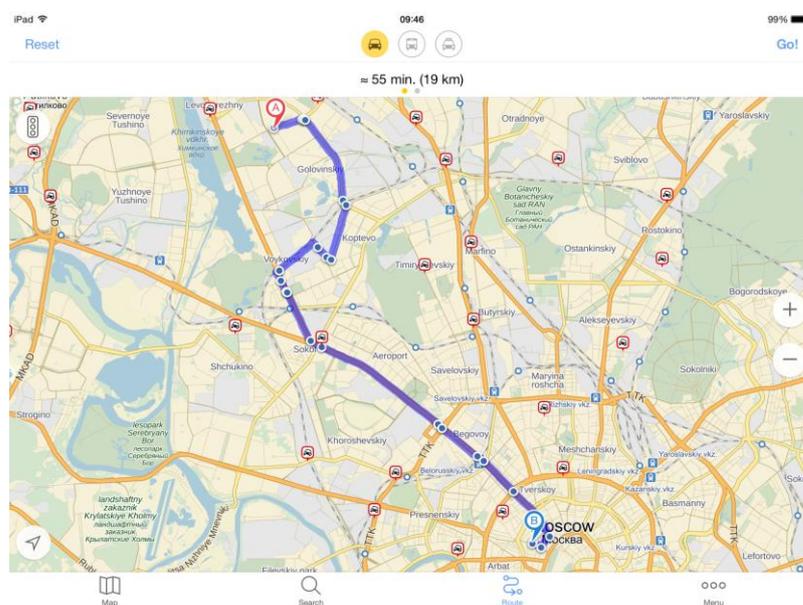


Рис. 7: Пример экрана системы Яндекс Карты (изображение траектории)

Мобильная навигационно-картографическая система **Яндекс Навигатор** [29] от той же компании Яндекс в отличие от Яндекс Карт ближе к навигационным решениям для автомобиля. Она имеет возможность описания всего маршрута только при помощи визуального представления траектории пути

(рис. 8). В качестве ввода использует точки начала, окончания маршрута и транзитные места. Прокладывание маршрута происходит в ароматическом режиме и не может быть изменено.

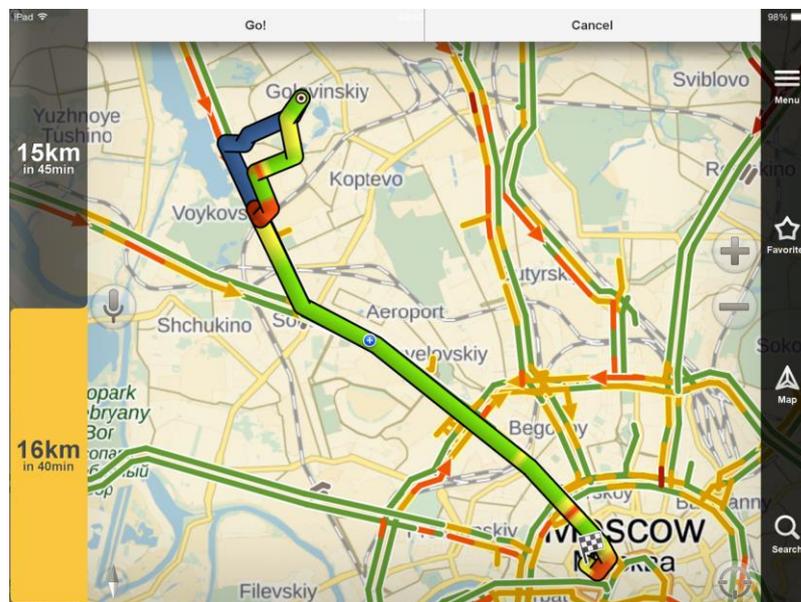


Рис. 8: Пример экрана системы Яндекс Навигатор (изображение траектории)

### 1.2.3 Носимые навигационные устройства

Смарт-часы и смарт-очки стали производиться сравнительно недавно, но уже на текущий момент могут помогать пользователю в навигации. Программные решения находятся в стадии глубокой разработки. Их рассмотрение на текущий момент не отражает итогового качества продукта. Сравнению подлежали платформы:

- Google Wear [30]
- Apple Watch [31]
- Google Glass [32]

Компьютерная система **Google Wear** (рис. 9) [30] производства компании Google работает в связке со смартфоном, передавая системные данные по каналу Bluetooth. Навигационно-картографическое приложение на смарт-часах подсоединяется к Google Maps и осуществляет своего рода вывод на

дополнительный экран, доступный прямо на запястье пользователя, когда сам смартфон находится в кармане или сумке. При начале навигации к точке, заданной при помощи голосовой команды с использованием системы распознавания речи Google Now [44], на экране смарт-часов отображается весь путь целиком в виде изображения траектории и спрашивается подтверждение пользователя о том, что он действительно желает начать по нему навигацию. Маршрут при этом прокладывается автоматически от текущего местоположения.



Рис. 9: Пример пользовательского интерфейса системы Google Wear

Смарт-часы **Apple Watch** (рис. 10) [31] производства компании Apple предлагают аналогичный функционал. За начальную точку пути принимается текущее местоположение пользователя, конечная точка задается через голосового ассистента Siri [43]. Маршрут прокладывается в автоматическом режиме без возможности его изменения. Перед началом навигации пользователю показывается путь целиком в виде изображения траектории на карте местности.



Рис. 10: Пример пользовательского интерфейса системы Apple Watch

Аналогичным в плане навигационно-картографического функционала являются смарт-очки **Google Glass** (рис. 11) [32] от Google. Устройство закрепляется на голове пользователя как обычные очки, с правой стороны расположен маленький проектор, проецирующий изображение на внутреннюю сторону правого глаза таким образом, что пользователь как-будто видит голограмму пользовательского интерфейса, висящего в воздухе на некотором расстоянии от лица. Приложение для осуществления навигации очень похоже на упомянутые выше приложения для смарт-часов. Тем же самым способом, что и в Google Wear, пользователь задает конечную точку маршрута при помощи голосового ассистента Google Now [44], за стартовую точку берется текущее местоположение, переданное на очки со смартфона, к которому они подсоединены по беспроводному каналу Bluetooth. Перед началом навигации на экране отображается изображение траектории всего маршрута.

Все рассмотренные системы не имеют возможности задавать тот маршрут, который интересует пользователя. То есть, возможен только автоматически проложенный путь от текущего местоположения до заданной голосом точки. Начальную точку задать не представляется возможным. Конечная точка задается,

исходя из заранее добавленных в «избранные» географических координат, к которым поставлены в соответствие некоторые названия, например, «дом» или «работа», либо при помощи голосового ввода названия POI, это может быть имя ресторана или магазина, станция метро, название улицы или адрес целиком, который будет обработан способом, аналогичным онлайн-навигационно-картографическим системам, предлагающим полосу для текстового поиска.



Рис. 11: Пример пользовательского интерфейса (того, что видит пользователь с надетым на голову устройством) системы Google Glass

#### **1.2.4 Стационарные навигационные устройства**

Навигационные стационарные терминалы (рис. 12) обычно располагаются в крупных зданиях, это могут быть музеи, аэропорты или торговые центры. Их задачей является четкое и краткое описание маршрута, которое сможет запомнить пользователь, так как взять с собой данный терминал не представляется возможным. Они, как правило, не имеют возможности вводить стартовую точку (за нее принимается текущее местоположение терминала) и в них нельзя вводить желаемый маршрут, что и не требуется. Главная задача такого вида устройств — четко и ясно описать маршрут до желаемого места пользователю так, чтобы он смог быстро понять его, запомнить и добраться до цели по памяти. Для решения этой задачи используется в основном изображение траектории, нанесенное на

план здания. План здания иногда делают трехмерным и дают пользователю поворачивать его. Это особенно актуально в ситуациях, когда проложенный маршрут проходит через несколько этажей.



Рис. 12: Стационарный терминал в торговом центре с сенсорным экраном

Не все стационарные терминалы обладают сенсорным экраном (рис. 13) и могут принимать ввод от пользователя. В таком случае задача по описанию маршрута ложится в руки дизайнеров карты. Необходимо таким образом нарисовать план и разметить на нем POI, чтобы у пользователя не возникло двоякого восприятия информации, все было четко и однозначно понятно. Подобные системы плохо работают в многоуровневых зданиях с большой площадью, где планировка разных этажей сильно отличается друг от друга.

Общая проблема, которая не может быть решена ни одним из рассмотренных видов стационарных терминалов — прокладывание сложных маршрутов между несколькими этажами. Например:

- маршрут с первого этажа в комнату на 4 этаже лежит через второй этаж, где нужно перейти в другую лифтовую шахту;

- маршрут с первого этажа в комнату на 2 этаже лежит через третий этаж (актуально для старых зданий, планировка которых менялась частями с течением времени).



Рис. 13: Стационарный терминал в торговом центре без сенсорного экрана

### 1.2.5 Онлайн-навигационные системы

Помимо рассмотренных выше отдельных устройств и программ для выполнения навигационно-картографических задач существует онлайн-навигационные системы, доступные непосредственно из веб-браузера и не требующие никаких установок локально. Подобные системы работают преимущественно на стационарных компьютерах, но могут быть также запущены на смартфонах и планшетах (данный вариант использования является сравнительно редким, так как для мобильных устройств существуют специализированные навигационно-картографические приложения, рассмотренные выше). Рассмотрению подлежали:

- Google Maps (рис. 14) [33]
- Яндекс Карты (рис. 15) [34]
- Bing Maps (рис. 16) [35]

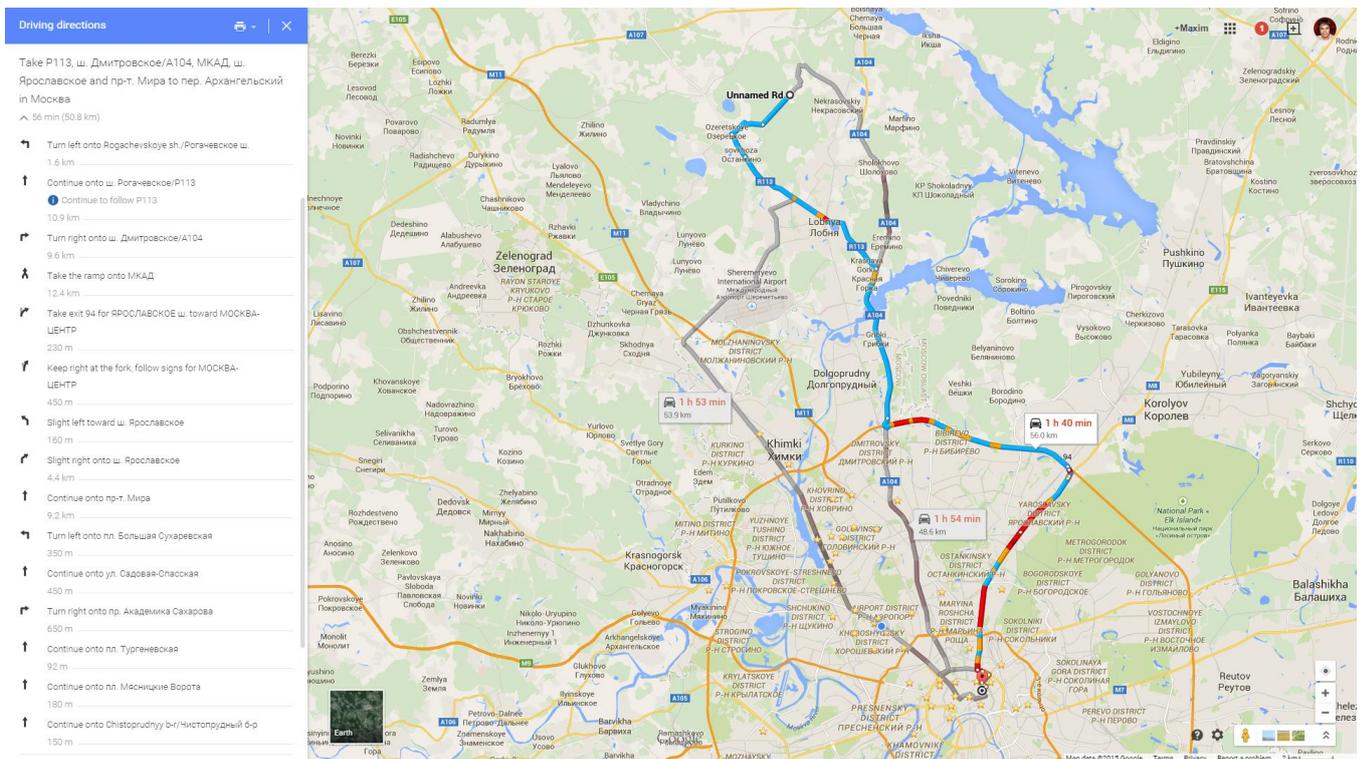


Рис. 14: Пример пользовательского интерфейса системы Google Maps

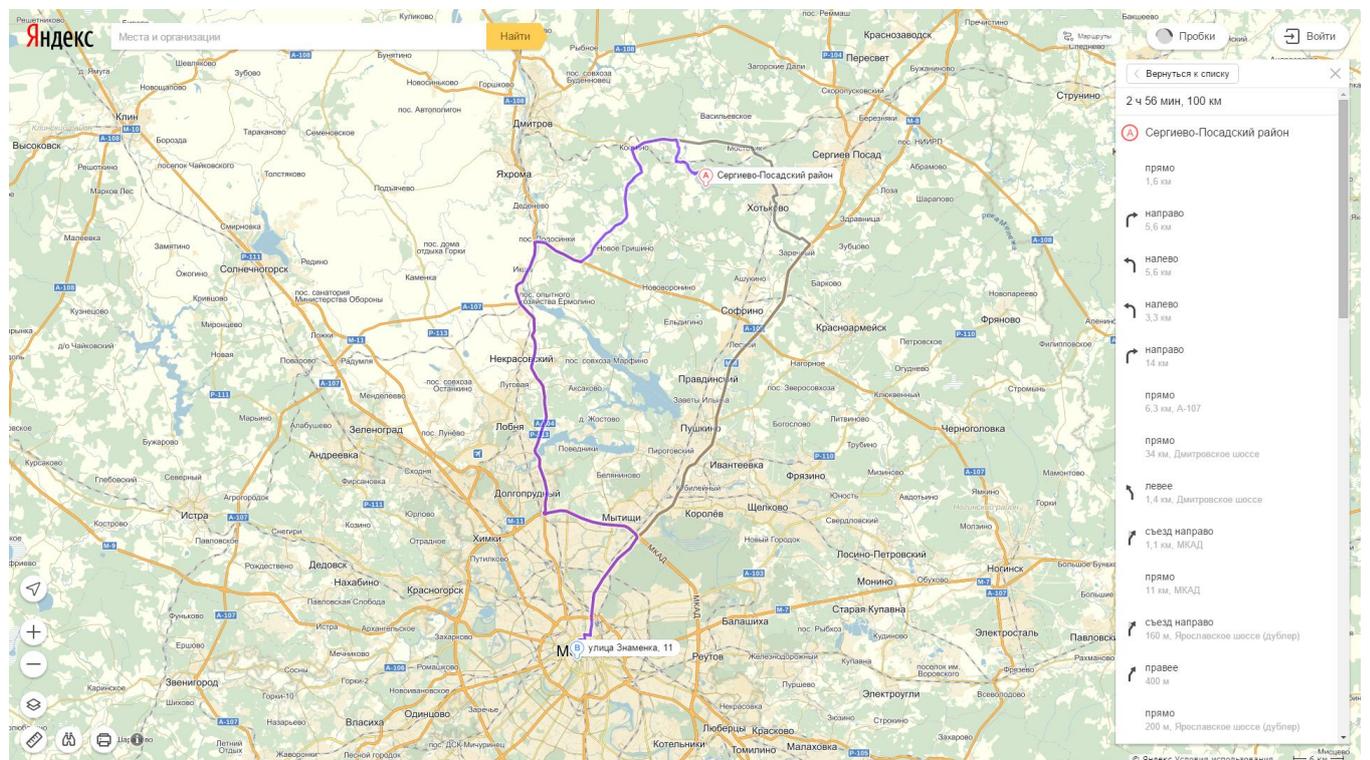


Рис. 15: Пример пользовательского интерфейса системы Яндекс Карты

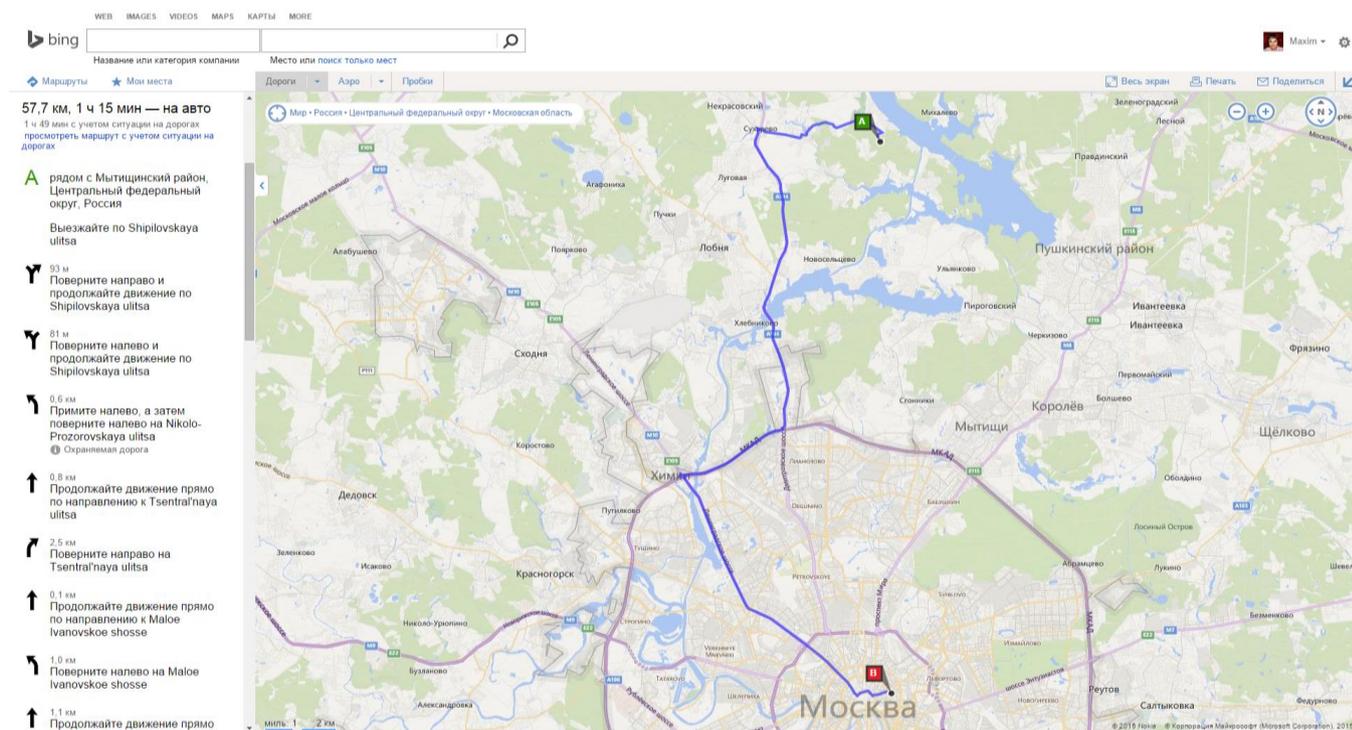


Рис. 16: Пример пользовательского интерфейса системы Bing Maps

Все три рассматриваемые онлайн-системы предлагают абсолютно идентичный функционал и отличаются преимущественно:

- внешним видом и удобством пользовательского интерфейса;
- картографической базой и списком известных системе POI, по которым можно осуществлять поиск.

Онлайн-системы предлагают описание маршрута в двух существующих вариантах: изображение траектории и подробная пошаговая инструкция к перемещению. Ввод маршрута возможен двумя способами:

- задание начальной и конечной точки (в качестве начальной точки может выступать, в том числе, текущее местоположение компьютера, определяемое либо по GPS, либо по данным от провайдера мобильного интернета, либо по геолокационным данным Wi-Fi точки доступа) и автоматическое прокладывание между ними маршрута;
- аналогичное предыдущему пункту задание начальной и конечной точек и изменение проложенного маршрута при помощи

перетаскивания опорных точек мышкой по карте так, что в итоге можно добиться желаемого маршрута (такой функционал доступен только на стационарных компьютерах и ноутбуках, использующих в качестве манипулятора компьютерную мышь).

### **1.3 Основные методы построения и распознавания описания маршрута**

Проведенный выше анализ навигационно-картографических систем позволяет выделить два существующих метода описания маршрута и два метода распознавания маршрута по описанию:

- методы построения описания маршрута:
  - пошаговая инструкция;
  - изображение траектории;
- методы распознавания описания маршрута:
  - ввод опорных точек маршрута путем их расстановки мышкой на стационарном компьютере; между ними маршрут прокладывается автоматически (данный способ позволяет максимально точно задать траекторию, при этом по-прежнему учитывает правила дорожного движения и никогда не прокладывает маршрут по дороге с односторонним движением в обратном направлении, что несомненно является плюсом);
  - ввод точек транзита путем задания их конкретных координат или имени POI (менее удобный с точки зрения уточнения сложных маршрутов; часть автомобильных навигаторов, где преимущественно используется метод добавления транзитных точек, поддерживают их ограниченное количество и не позволяют полностью описать желаемый маршрут).

Далее более детально рассматривается каждый метод построения и распознавания описания маршрута с учетом всех выявленных минусов и плюсов.

### 1.3.1 Пошаговая инструкция

Пошаговая инструкция представляет собой последовательность действий (рис. 17), строгое очередное выполнение которых приведет пользователя из стартовой точки в финишную.

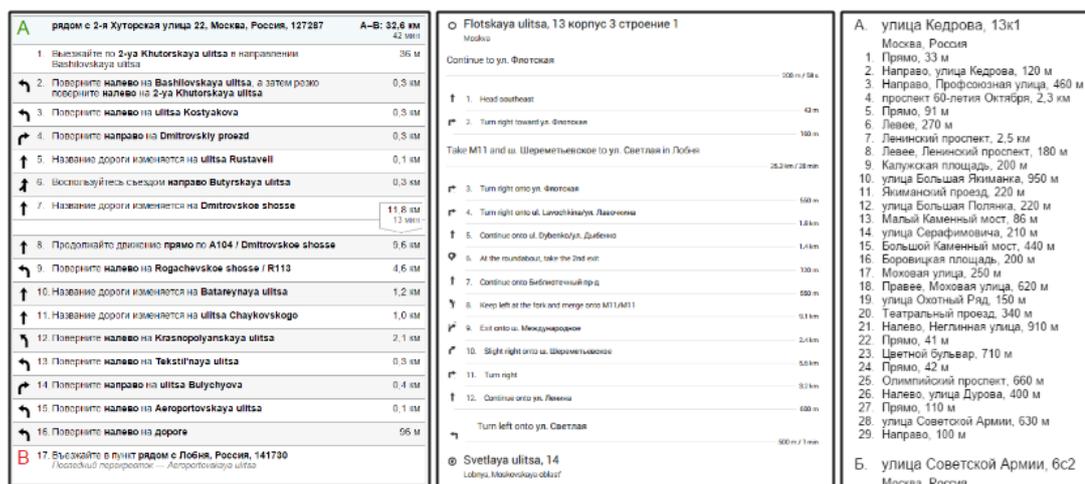


Рис. 17: Описание маршрута в виде пошаговой инструкции к перемещению в навигационно-картографических системах Bing Maps, Google Maps, Яндекс Карт (слева направо)

Данный метод описания пути позволяет:

- узнать точную последовательность действий, приводящую из точки старта в точку окончания пути;
- понять приблизительную сложность маршрута, исходя из количества инструкций;
- приблизительно оценить общую продолжительность движения по маршруту, исходя из количества инструкций;
- узнать мелкие детали, даже на очень длинном маршруте.

При этом, пошаговая инструкция не позволяет:

- точно определить свое местоположение на пути;
- понять общее направление движения;
- легко запомнить инструкцию целиком и проехать по пути по памяти;
- при движении сориентироваться относительно хорошо заметных объектов на пути из-за отсутствия сведений о них в описании;
- сориентироваться в расположении маршрута относительно крупных для него объектов: городов, улиц, зданий.

### 1.3.2 Изображение траектории

Изображение траектории (рис. 18) представляет собой вид на карту сверху с выделенной ломаной линией маршрута, соединяющей стартовую и финишную точки. Чем длиннее маршрут, тем менее детальным становится представление карты из-за ограничений отображающих устройств.

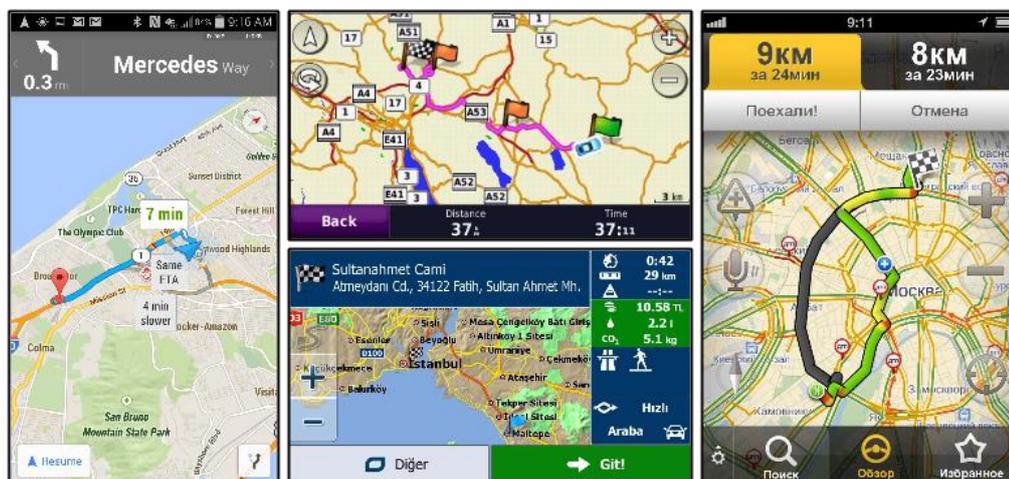


Рис. 18: Описание маршрута в виде изображения траектории пути в навигационно-картографических системах Google Maps, Garmin, iGo, Яндекс Навигатор (слева направо, сверху вниз)

Таким образом, исходя из проведенного анализа систем [23-35], изображение траектории позволяет:

- сориентироваться в расположении маршрута относительно крупных для него объектов: городов, улиц, зданий;
- понять общее направление движения и приблизительную длину пути.

При этом, изображение траектории не позволяет:

- использовать данный метод при продолжительном движении по незнакомой местности, так как при длинных маршрутах пропадают мелкие детали пути, делающие данный метод бесполезным при движении по незнакомой местности;
- сориентироваться относительно хорошо заметных объектов на пути из-за отсутствия сведений о них;
- узнать точную информацию о проезжаемых объектах, улицах и т.д.;
- запомнить длинный маршрут и проехать по нему по памяти.

### **1.3.3 Использование опорных точек**

Для задания конкретного маршрута по результатам анализа систем [23-35] были выявлены два способа, похожие друг на друга:

- задание стартовой и конечной точки маршрута и добавление точек транзита, через которые он должен проходить; маршрут между всеми точками прокладывается автоматически, количество транзитных точек ограничено и тем самым не любой маршрут может быть задан точно так, как того желает пользователь;
- задание стартовой и конечной точки и корректировка маршрута при помощи перемещения его частей в нужные места с использованием компьютерного манипулятора мыши; добавление дополнительных точек не ограничено их числом, маршрут между всеми точками прокладывается автоматически.

Таким образом, для этих двух вариантов можно выделить общее название — описание маршрута с использованием опорных точек (рис. 19). Первым минусом данного подхода является необходимость добавления новых опорных

точек в описание, если в автоматическом режиме маршрут был проложен не в соответствии с желанием пользователя. Вторым минусом является строгое требование к аппаратному обеспечению и наличию графического интерфейса. Третий минус — невозможность использовать знакомые конкретно данному пользователю объекты и части маршрута, по которым он часто перемещается. Иными словами, отсутствует персонализация вводимых данных.

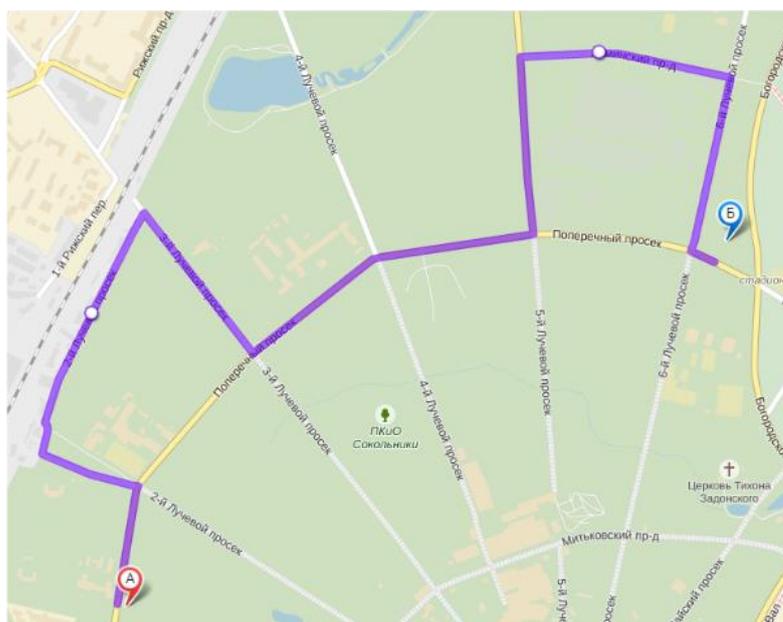


Рис. 19: Пример введенного маршрута при помощи задания опорных точек в онлайн сервисе Яндекс Карты

Современные устройства поддерживают голосовой ввод информации. Новейшие платформы, такие как смарт-очки и смарт-часы, часто полагаются только на голосовой ввод. Средств описания маршрута голосом, чтобы его могла воспринять система, пока либо не существует, либо об их разработке пока не заявлено публично. В любом случае, задать конкретный маршрут голосом на текущий момент на широко распространенных мобильных платформах невозможно.

## **1.4 Выводы к первой главе и выбор направления исследования**

На основе проведенного в данной главе анализа существующих навигационно-картографических систем и применяемых в них способах построения и распознавания описания маршрутов можно сделать следующие выводы:

- существующие на сегодняшний день в навигационно-картографических системах методы построения описания маршрута не обладают одновременно удобством для восприятия и запоминания, универсальностью использования на разных устройствах (обязательно требуют экран) и подстройкой под знания конкретного пользователя, не предоставляя тем самым возможность осуществления когнитивной навигации;
- существующие на сегодняшний день в навигационно-картографических системах методы распознавания описания маршрута не универсальны (не могут использоваться на разных платформах, требовательны к оборудованию пользователя) и не обладают персонализацией, тем самым не позволяют использовать в описании знакомые конкретному пользователю объекты и интервалы пути.

Универсальным форматом ввода и вывода маршрута является его текстовое словесное описание, исходя из фактов:

- текстовое представление может быть синтезировано в речь и использоваться в роботизированных системах, у которых отсутствует экран;
- текстовое представление может быть отображено на любом экране (черно-белом, цветном и даже только текстовом);
- текстовое представление с учетом развития технологий распознавания речи и персональных ассистентов, таких как Siri и

Google Now, работающих исключительно на голосовых командах, облегчает человеку ввод данных в компьютер;

- текстовое представление описания, имеющее вид прямой речи человека (т.е. не формальный строгий формат, а свободный разговорный стиль), является легким для восприятия и запоминания, так как использует привычные для пользователя выражения и обороты из повседневной жизни.

Выводы об удобстве использования человеком текстового описания маршрута были получены на основе проведенных исследований совместно с факультетом психологии МГУ им. М.В. Ломоносова с использованием специальной системы построения виртуальной реальности CAVE (рис. 20) [1, 2].

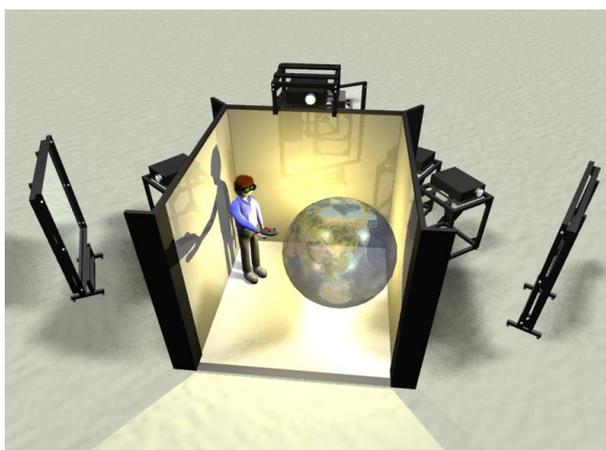


Рис. 20: Схематическое представление системы построения виртуальной реальности CAVE

Таким образом, на основе проведенных анализов и сделанных выводов было выбрано направление исследования и поставлена научная задача по разработке методов построения персонализированных текстовых навигационных описаний маршрутов для картографических компьютерных систем, позволяющих упростить взаимодействие человека с компьютерными и роботизированными системами, а также их реализация в виде программного модуля, подключаемого к существующим навигационно-картографическим сервисам.

## Глава 2. Предложенные методы

В данной главе рассматриваются предлагаемые методы построения и распознавания текстового описания маршрута. Помимо этого, затрагиваются проблемы персонализации, чьей задачей является подстройка поведения алгоритмов под конкретного пользователя и использование знакомых лично ему мест и маршрутов на карте, проблемы упрощения получаемого текста с целью придания «человечности» формулировкам и проблемы склонения названий. Кроме того, решаются вопросы, связанные с неточностью введенной пользователем информации и наличием нескольких подходящих под описание маршрутов.

### 2.1 Построение текстового описания маршрута

В общем представлении последовательность действий для построения текстового описания маршрута состоит из следующих шагов [8]:

1. получение от пользователя названий или координат точек старта и конца пути;
2. построение маршрута между заданными точками в автоматическом режиме с использованием существующих навигационно-картографических онлайн-систем;
3. получение от навигационно-картографической системы подробного описания маршрута с конкретными директивами к перемещению;
4. работа алгоритма построения текстового описания на основе полученных данных;
5. отображение результата пользователю.

Все пункты, кроме 4-го, имеют большее отношение к программной реализации, описанной в следующей главе. Далее по тексту детально разобран пункт 4.

В описании алгоритма часто приводятся примеры различных лексем на русском языке, которые используются в финальном текстовом представлении, например, «вперед» или «полкилометра».

### 2.1.1 Описание алгоритма

Алгоритм базируется на входных данных определенного содержания, состоит из последовательных шагов их преобразования в итоговое текстовое описание. Данные, получаемые на вход от навигационно-картографической системы, представляют собой тройку  $In = \{L, M, D\}$ :

1.  $L$  — полигон маршрута;
  - а. представляет собой массив двоек  $\{Latitude, Longitude\}$  вершин ломаной линии, описывающей траекторию маршрута в реальном мире;
2.  $M$  — текстовые метки с названиями улиц;
  - а. представляют собой массив троек  $\{Latitude, Longitude, Name\}$ , в котором хранится информация о проезжаемых пользователем улицах;
3.  $D$  — навигационные директивы;
  - а. представляют собой массив троек  $\{Latitude, Longitude, Directive\}$ , в котором хранится информация о последовательных действиях, которые необходимо предпринять для достижения финишной точки вида «прямо», «направо», «разворот» и другие.

Помимо входных данных от навигационно-картографической системы используются дополнительные источники данных, хранящиеся локально:

1.  $NL$  — знакомые пользователю маршруты;
  - а. представляет собой массив из *именованных* массивов двоек  $\{Latitude, Longitude\}$ , являющихся знакомыми пользователю маршрутами или их частями, то есть теми интервалами

пути, по которым он часто перемещается и может выделить для них особое название, например, «далее, как на работу»;

2. *NP* — знакомые пользователю POI;

а. представляет собой массив четверок  $\{Latitude, Longitude, Name, Popularity\}$ , в котором хранится информация о местах, в которых часто бывал пользователь, в поле *Popularity* заносится общее число визитов;

Все хранимые локально данные имеют собственный уникальный идентификатор, выражаемый целым положительным числом. Благодаря использованию полностью уникального идентификатора появляется возможность использовать адресацию к информации по нему без указания типа самих данных. Здесь и далее идентификатор обозначается как *ID*.

В ходе работы алгоритма могут потребоваться дополнительные данные, запрашиваемые от навигационно-картографической системы:

- *RP* — набор POI в некотором радиусе от заданной точки, запрашивается при необходимости дополнить текстовое описание;
- представляет собой массив четверок  $\{Latitude, Longitude, Name, Popularity\}$ , в котором хранится информация об общественных местах (магазинах, организациях, парках, остановках, достопримечательностях и т.д.), в поле *Popularity* хранится обобщенная оценка места, основанная на пользовательских комментариях;

В алгоритме используется понятие *каркаса* — набор лексем, объединенных в последовательность на основе входных данных, после чего заменяемых на слово или словосочетание на русском языке согласно определенным для каждой лексемы правилам. Каркас в общем случае обозначается греческом буквой  $\Sigma$ .

Высокоуровневое представление шагов работы алгоритма (рис. 21):

1. выделение из полигона маршрута, полученного от навигационно-картографической системы, знакомых пользователю интервалов;
2. фиксация близлежащих к маршруту знакомых пользователю POI в качестве ориентиров;
3. группировка интервалов по однотипным навигационным директивам;
4. формирование каркаса результата из лексем;
5. склонение названий и интеграция их в каркас;
6. замена оставшихся лексем по словарю;
7. загрузка изображений POI;
8. формирование результирующего текстового описания в формате HTML с изображениями и с визуальным форматированием.

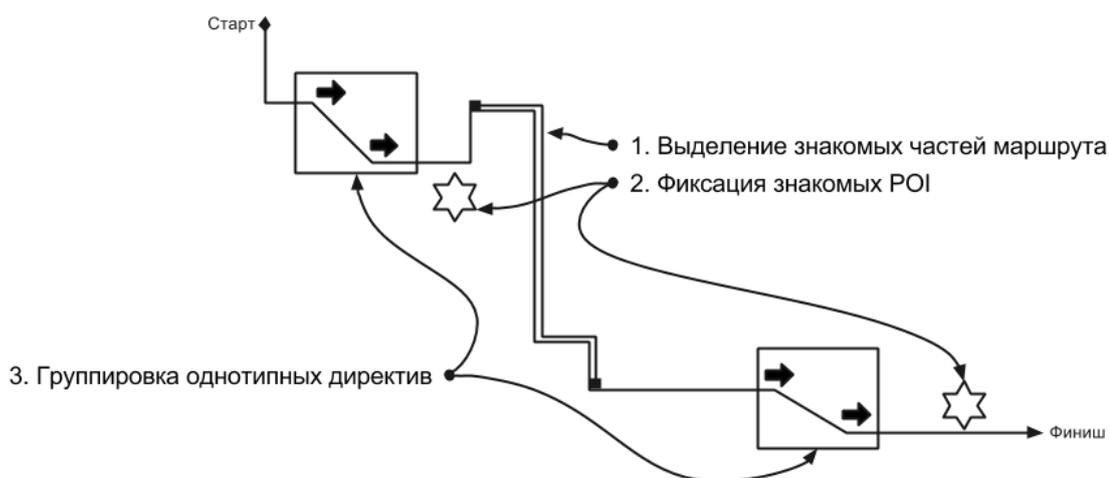


Рис. 21: Визуальное представление первых трех шагов на схеме маршрута [16]

### **Выделение знакомых пользователю интервалов из полигона маршрута.**

Алгоритм использует локально хранимые данные по ранее пройденным пользователем маршрутам  $NL$ . Для выделения из  $L$  знакомых пользователю интервалов, имеющих в хранилище  $NL$ , производится анализ полного или частичного совпадения этих путей с частями полигона. Таким образом, производится поиск таких  $L'$ :

$$L = (p_n)_{n=1}^m, p_n = \{lat_n, lng_n\}, m > 0;$$

$$NL = (q_n)_{n=1}^k, q_n = \{lat_n, lng_n\}, k > 0;$$

$$L^c = S: \exists l > 0, \exists t > 0, l \leq t,$$

$$S \subseteq NL: \forall l < l', (q_n)_{n=1}^k \neq (p_n)_{n=t}^{l'+t}, \forall l' > l, (q_n)_{n=1}^{l'} = (p_n)_{n=t}^{l'+t}.$$

Представим соответствующую логику поиска  $L^c$  в виде псевдокода, близкого по синтаксису к JavaScript:

```
var foundSubPaths = {};
for (var i = 0; i < knownPaths.length; i++) {
    var lastKnownPoint = knownPaths[i].points[knownPaths[i].points.length - 1];
    var belongsToCurrentPath = currentPath.contains(lastKnownPoint);
    if (belongsToCurrentPath) {
        for (var j = knownPaths[i].points.length - 2; j >= 0; j--) {
            var knownPoint = knownPaths[i].points[j];
            var belongsToCurrentPath = currentPath.contains(knownPoint);
            if (belongsToCurrentPath) {
                foundSubPaths.append(knownPaths[i], knownPoint);
            }
        }
    }
}
}
```

Таким образом, в представленном псевдокоде реализуется поиск среди всех знакомых пользователю маршрутов тех, чьи конечные точки принадлежат обрабатываемому полигону. Далее для каждого такого маршрута в цикле находится самая первая точка, с которой началось совпадение. В листинге не указано, но в алгоритме также учитываются следующие параметры и аспекты:

- $NL\delta$  — коэффициент принадлежности точки маршруту, то есть какое минимальное расстояние должно быть между точкой и ломаной линией полигона, чтобы считать ее принадлежащей маршруту;
  - проблема возникает тогда, когда полигон  $L$  содержит в себе конец любого полигона из  $NL$ , но записанный немного смещенными в пространстве географическими координатами; такое возникает из-за сильных шумовых отклонений систем GPS и ГЛОНАСС;
  - расстояние между точкой и полигоном считается следующим образом: находятся две ближайшие точки полигона к той точке, чья принадлежность проверяется; далее от нее рассчитывается расстояние до прямой между найденными точками; если расстояние меньше заданного коэффициента  $NL\delta$ , то точка считается принадлежащей.
- $NL\rho$  — минимальное расстояние, которое должен занимать найденный маршрут в метрах, то есть случаи, когда общий интервал между знакомым пользователем маршрутом и полигоном меньше некоторого порогового значения, пропускаются;
- маршруты, которые входят в полигон разными частями (началом, серединой, концом) несколько раз, алгоритмом будут задействованы только единожды на конечном интервале (так как знакомые пользователю маршруты из  $NL$  имеют названия с ссылкой на конечный пункт, например, «на работу», соответственно могут быть использованы только в ситуациях, когда выражение «и далее, как на работу» является уместным);

**Фиксация знакомых пользователю POI в качестве ориентиров** происходит схожим методом. Находятся такие  $P^*$  из множества хранящихся в

локальной базе  $NP$ , которые находятся поблизости от маршрута  $L$ , близость определяется параметром  $NP\rho$ :

$$L = (p_n)_{n=1}^m, p_n = \{lat_n, lng_n\}, m > 0;$$

$$NP = q_n, q_n = \{lat_n, lng_n\};$$

$$P^c: P^c \subseteq NP, dist(P^c, L) < NP\rho.$$

Реализация поиска  $P^c$  в виде псевдокода, близкого по синтаксису к JavaScript, следующая:

```
var foundPOIs = [];  
for (var i = 0; i < POIs.length; i++) {  
    var containsPOI = currentPath.contains(POIs[i].location);  
    if (containsPOI) {  
        foundPOIs.append(POIs[i]);  
    }  
}
```

Таким образом, получается список всех POI, которые находятся рядом с маршрутом, при этом в листинге не указано, но в алгоритме учитывается:

- $NP\rho$  — минимальное расстояние, которое может быть от полигона до POI, рассчитывается аналогично описанному выше методом по определению принадлежности точки полигону маршрута;

Далее происходит фильтрация всех найденных POI и выделение из них наиболее знакомых пользователю или, если таких нет, наиболее известных с точки зрения общественного мнения, выраженного в оценке POI, хранящейся в навигационно-картографической системе (осуществляется дополнительный запрос на получение  $RP$ ). Для этого:

- для каждого логического интервала (между навигационными директивами) полигона выбирается одна POI с наибольшим рейтингом известности пользователю или, если его нет ни у одного найденного POI, с наибольшим общественным рейтингом;
  - данные POI будут впоследствии участвовать в формулировках вида «проезжайте мимо ...».
- для каждого окончания интервала (рядом с географическим местом, где необходимо выполнить навигационную директиву, то есть на поворотах, разворотах и т.д.) аналогичным образом выбирается по одной POI.
  - данные POI будут впоследствии участвовать в формулировках вида «поверните рядом ...».

**Группировка интервалов по однотипным навигационным директивам** происходит следующим образом:

1. все директивы «прямо» группируются в одну, преодолеваемое расстояние суммируется (расстояние вычисляется исходя из полигона, описывающего конкретный интервал пути);
2. все директивы поворотов, расстояние между которыми меньше заданного настройками порога (предполагается, что это очень близкие друг к другу повороты), группируются в одну с указанием их числа.

Подобные группировки позволяют сократить итоговое текстовое описание, тем самым упростив его и сделав более легким для восприятия и запоминания.

**Формирование каркаса результата из лексем** является одним из самых важных шагов алгоритма. На этом этапе строится последовательность лексем, которая в итоге будет преобразована в текст. При построении каркаса

используются следующие формы, представимые в виде одной или нескольких лексем [9]:

- полные и упрощенные названия POI, используемые в прямой речи;
- словесное описание внешнего вида POI;
- склонения названий POI (`{POI_NAME}`);
- знакомые пользователю интервалы пути (`{KNOWN_NAME}`);
- директивы к перемещению (`{ACTION}`, `{DIRECTION}`);
- упрощенные округленные расстояния (`{GROUP_DISTANCE}`);
- группы однотипных действий (`{GROUP_DIRECTION}`);
- предлоги, союзы, знаки препинания (`{TILL}`, «,»);
- «мусорные» слова, придающие тексту «человечности» (`{JUNK}`).

Для построения каркаса используется формальная грамматика — способ описания формального языка, то есть выделение некоторого подмножества из множества всех слов некоторого конечного алфавита. Различают порождающие и распознающие (или аналитические) грамматики — первые задают правила, с помощью которых можно построить любое слово языка, а вторые позволяют по данному слову определить, входит ли оно в язык или нет. Для задачи построения каркаса используется порождающая грамматика, определяемая следующими характеристиками:

- E — набор (алфавит) терминальных символов;
- N — набор (алфавит) нетерминальных символов;
- P — набор правил вида: «левая часть» → «правая часть», где:
- «левая часть» — непустая последовательность терминалов и нетерминалов, содержащая хотя бы один нетерминал;
- «правая часть» — любая последовательность терминалов и нетерминалов;
- S — стартовый (или начальный символ) грамматики из набора нетерминалов.

Терминал (терминальный символ) — объект, непосредственно присутствующий в словах языка, соответствующего грамматике, и имеющий конкретное, неизменяемое значение (обобщение понятия «буквы»). Нетерминал (нетерминальный символ) — объект, обозначающий какую-либо сущность языка (пример: формула, арифметическое выражение, команда) и не имеющий конкретного символьного значения.

В рамках текущей реализации следующие наименования считаются синонимами:

- терминальный символ = лексема;
- слово = каркас;

Для реализации алгоритма используется следующая формальная грамматика:

- E:
  - {NULL}, {JUNK}, {ACTION}, {DIRECTION}, {DISTANCE}, {NEAR}, {TILL}, {POI\_NAME}, {KOWN\_ACTION}, {KNOWN\_NAME}, {GROUP\_DIRECTION}, {GROUP\_DISTANCE};
- N:
  - S, SE, SP, JUNK, ACT, DIR, DIST, NEARPOI, TILLPOI, NACTDIR, GDIR, GDIST, NEAR, TILL, POI;
- S: S.

Для определения набора правил P использованы некоторые обобщения, позволяющие сократить описание при сохранении основного смысла:

- тремя точками обозначаются длинные последовательности из одинаковых элементов, количество элементов определяется общим числом навигационных директив в формальном описании маршрута, полученном от навигационно-картографической системы;

- вертикальной чертой или несколькими правилами с одинаковой левой частью обозначаются возможные варианты; способ вычисления вероятности выбора одного или другого варианта описан ниже по тексту.

Набор правил  $P$  определяется следующим образом:

```

S → SE. SE. ...
SE → SP, SP, ...
SP → JUNK ACT DIR DIST NEARPOI TILLPOI
SP → JUNK NACTDIR
SP → JUNK ACT GDIR GDIST NEARPOI TILLPOI
JUNK → {JUNK} | {NULL}
ACT → {ACTION} | {NULL}
DIR → {DIRECTION} | {NULL}
DIST → {DISTANCE} | {NULL}
NEARPOI → NEAR POI
NEARPOI → {NULL}
TILLPOI → TILL POI
TILLPOI → {NULL}
NEAR → {NEAR}
TILL → {TILL}
POI → {POI_NAME}
NACTDIR → {KNOWN_ACTION} {KNOWN_NAME}
GDIR → {GROUP_DIRECTION}
GDIST → {DROUP_DISTANCE}

```

В качестве примера приведем следующий каркас, который будет использоваться на следующих шагах алгоритма для иллюстрации производимых действий:

```

{ACTION} {DIRECTION} {DISTANCE}, {JUNK} {TURN_LEFT}, {JUNK} {ACTION}
  {GROUP_DIRECTION} {GROUP_DISTANCE} {TILL} {POI_NAME}, {JUNK}
{KNOWN_ACTION} {KNOWN_NAME}, {JUNK} {ACTION} {GROUP_DIRECTION}
  {GROUP_DISTANCE} {TILL} {POI_NAME}.

```

**Склонение названий и интеграция их в каркас** представляет собой сравнительно простой шаг алгоритма, выполняющий последовательно два действия:

- склонение используемых названий POI (для лексем {POI\_NAME} получается из локального хранилища *NP*);
- замена лексем {POI\_NAME} и {KNOWN\_NAME} на соответствующие им словесные формы;

Основную сложность представляет алгоритм склонения. Ниже в главе, освещающей программную реализацию, рассказано о возникших сложностях и методах их решения. Текстовые формы для лексем {KNOWN\_NAME} не подлежат склонению, так как заранее на момент сохранения уже имеют требуемое представление в *NL*.

Таким образом, в упомянутом выше примере произойдут очевидные изменения, приводящие его к виду:

```
{ACTION} {DIRECTION} {DISTANCE}, {JUNK} {TURN_LEFT}, {JUNK} {ACTION}
{GROUP_DIRECTION} {GROUP_DISTANCE} {TILL} Большого театра, {JUNK}
{KNOWN_ACTION} с работы домой, {JUNK} {ACTION} {GROUP_DIRECTION}
{GROUP_DISTANCE} {TILL} Библиотеки им. Ленина.
```

**Замена оставшихся лексем по словарю** происходит в самом конце. Задачей данного шага является подстановка слов на русском языке в нужном склонении. Все формы слов хранятся в таблице и не требуют преобразований на этапе выполнения алгоритма:

Проезжайте прямо четверть километра, затем поверните налево, после чего двигайтесь прямо до **Большого театра**, далее как обычно **с работы домой**, и прямо километр до **Библиотеки им. Ленина**.

**Загрузка изображений POI** представляет собой получение URL ссылок на картографический сервис, по которой располагаются изображения POI с нужной позиции под заданным ракурсом (рис. 22). В случае навигации по городу обязательным условием является нахождение точки, с которой произведена фотография, на проезжей части дороги. Это связано с тем, что фотографии улиц делаются автомобилем, оснащенный панорамной камерой на крыше — фотографии могут быть только с тех позиций, где мог проехать автомобиль. В случае, если запрошена фотография с позиции, где не производилась съемка, будет использоваться ближайшая к ней доступная точка с панорамной фотографией.



Рис. 22: Поиск нужной точки на пути для получения фотографии POI от навигационно-картографического сервиса

Для формирования URL ссылки необходимо вычислить:

- географические координаты точки с видом, откуда требуется изображение POI;
- угол поворота камеры.

Географические координаты POI заранее известны. Требуется получить позицию точки, с которой требуется снимок, и угол поворота камеры. Для этого требуется выполнить следующие действия:

- определить ближайшую точку кривой пути к POI;
- отсчитать 100 метров по ней в сторону старта;
  - если это невозможно (POI находится близко к началу интервала), то отсчитать 10% в сторону старта на части интервала от его начала до ее ближайшей к POI точки;
- рассчитать угол поворота камеры, исходя из позиции, откуда делается снимок, и географических координат POI;

Результат формируется в URL запрос по правилам, определенным навигационно-картографической системой.

**Формирование результирующего текстового описания в формате HTML с изображениями и визуальным форматированием** происходит на последней стадии работы алгоритма. Выполняются последовательно действия:

- все лексемы, содержащие названия POI (`{POI_NAME}`), обрамляются в специальный тэг-блок, при наведении на который (или нажатии на который при выводе на сенсорный экран) показывается всплывающее окно с подробной информацией об объекте:
  - официальное название POI;
  - текстовое описание внешнего вида;
  - изображение POI с той стороны, с которой увидит его пользователь.
- итоговый текст в формате HTML выводится на экран.

Визуально весь алгоритм построения текстового описания маршрута может быть представлен следующим образом в виде блок-схемы, изображенной на рис. 23.



Рис. 23: Схема работы всего алгоритма

### 2.1.2 Хранилище данных, словарей и объектов

В данном параграфе более детально описаны хранимые локально данные и формат их представления [7]. Для алгоритма требуются:

1. хранилище персональных данных и дополнительной картографической информации об объектах (описанный в алгоритме как *NP*);
2. хранилище знакомых маршрутов (описанный в алгоритме как *NL*);
3. словарь навигационных терминов;
4. словарь дополнительных терминов;

**Хранилище персональных данных и дополнительной картографической информации об объектах** является структурой данных со следующими полями:

- *ID* — идентификатор записи;
- *PLACE\_ID* — идентификатор места на карте, используемой навигационно-картографической системой, является обычно

HEX-строкой, используется для получения дополнительной информации по POI в случае ее востребованности;

- *LAT\_LNG* — структура данных, хранящая долготу и широту местоположения объекта;
- *NAME* — название объекта, если оно применимо (имя учреждения, магазина и т.д.);
- *VISUAL\_DESCRIPTION* — визуальное описание объекта, прочитав которое пользователь может понять, о каком объекте идет речь, когда встретит его на пути;
- *SHORT\_DESCRIPTION* — короткое описание, которое используется непосредственно в текстовом описании маршрута для сокращения длины текста, а также в случае отсутствия имени у объекта;
- *COMMON\_POPULARITY* — значение популярности данного объекта, исходя из общественного мнения (обычно оценка от 1 до 10 в звездочках);
- *PRIVATE\_POPULARITY* — значение популярности данного объекта, исходя из личного мнения пользователя, вычисляется автоматически (при посещении или нахождении рядом) или указывается вручную; наличие значения личной популярности отображает факт знания пользователя о данном объекте;
- *EMOTIONAL* — эмоциональная составляющая, например, пользователь может часто посещать одно и то же место, которое ему абсолютно не нравится; указав в ручном режиме низкую эмоциональную оценку объекта, он больше не будет получать его в описаниях маршрута, и наоборот, объекты с высокой эмоциональной оценкой будут чаще использоваться алгоритмом.
- *ATTRIBUTES* — атрибуты объекта: *PASS\_THROUGH* (объект стоит на дороге и сквозь него можно пройти, например, арка), *NON\_REACHABLE* (объект, который может служить ориентиром, но

до которого нельзя добраться обычным способом, например, маяк в море).

**Хранилище знакомых маршрутов** является набором массивов географических координат, описывающих ломаные линии маршрутов, по которым когда-либо перемещался пользователь. Каждый маршрут имеет счетчик количества его прохождений *USAGE\_COUNTER*. Один маршрут отделяется от другого продолжительной паузой, означающей остановку в движении пользователя. Равенство маршрутов проверяется при помощи расчета суммарного отклонения ломаной линии одного маршрута от ломаной линии другого маршрута. У каждого маршрута можно задать имя *ROUTE\_NAME* и описание *ROUTE\_DESCRIPTION*, которые будут использоваться алгоритмом. Аналогично с POI у каждого маршрута есть параметр *EMOTIONAL*, хранящий его эмоциональную оценку. Он вводится в ручном режиме. Это позволяет чаще или реже использовать упоминание о маршруте при работе алгоритма.

**Словарь навигационных терминов** — набор слов на родном языке пользователя совместно с набором синонимов:

- *FORWARD* (например, «вперед»);
- *BACKWARD* (например, «назад»);
- *LEFT* (например, «налево»);
- *RIGHT* (например, «направо»);
- *UTURN* (например, «развернуться»);
- *TILL* (например, «до»);
- *NEAR* (например, «мимо»);
- *THROUGH* (например, «сквозь»);
- *WALK* (например, «идти»).

**Словарь дополнительных терминов** — слова, которые придают навигационному тексту подобие человеческой речи, призваны упростить восприятие и облегчить запоминание. Они подставляются на место лексемы {JUNK}, используются случайным образом при составлении описания. Количество слов может быть любым, например: «далее», «затем», «после этого», «потом».

### 2.1.3 Вычисление расстояний между точками на планете

Для вычисления расстояния между двумя точками на поверхности планеты, которые заданы парами  $\{Latitude, Longitude\}$ , недостаточно посчитать расстояния по теореме Пифагора, так как необходимо учитывать сферическую природу планеты и, соответственно, ее радиус [36]. Исходя из этого, в общем виде вычисление расстояния выглядит следующим образом:

$$a = \sin^2(\Delta\varphi / 2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda / 2),$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}),$$

$$d = R \cdot c.$$

Где,  $R$  — радиус планеты (округленное до трех знаков после запятой значение в 6.371 километр),  $d$  — искомое расстояние.

На псевдокоде, близком в JavaScript, реализация выглядит следующим образом:

```
var R = 6371000; // радиус планеты в метрах
var φ1 = lat1.toRadians();
var φ2 = lat2.toRadians();
var Δφ = (lat2-lat1).toRadians();
var Δλ = (lon2-lon1).toRadians();
```

```
var a = Math.sin(Δφ/2) * Math.sin(Δφ/2) + Math.cos(φ1) * Math.cos(φ2) *  
    Math.sin(Δλ/2) * Math.sin(Δλ/2);  
var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));  
var d = R * c;
```

#### **2.1.4 Проблема использования знакомых пользователю мест и путей**

Для использования знакомых пользователю мест и путей необходимо собирать о нем информацию:

- пол;
- возраст;
- вкусовые предпочтения;
- основные места пребывания;
- и другие.

Получение таких данных возможно из социальных сетей, либо непосредственно при помощи их ввода самим пользователем в интерфейсе программы [3].

Сбор персонализированной информации (о маршрутах и посещаемых местах) происходит при непосредственном перемещении пользователя по миру. Делается автоматическая запись координат положения мобильного телефона, их последующий анализ и запись в структуру хранения данных следующим образом:

- отслеживаются промежутки времени, при которых телефон был относительно неподвижен (остановки на более чем на 30 минут);
- отслеживаются быстрые (движение на автомобиле или в общественном транспорте) и медленные (перемещение пешком, при скорости менее 20 км/ч) изменения координат;
- при остановках возможно дополнительное уточнение у пользователя, какое конкретно место (заведение, фирму, магазин) он посещает (аналогично сделано в социальной сети «foursquare»).

Список доступных для посещения мест доступен благодаря описанным выше картографическим сервисам, предлагающим доступ к накопленной информации.

Дополнительно в структуру данных когнитивной карты заносится следующая информация:

- количество перемещений по конкретному маршруту или его участку;
- количество посещения различных POI;
- эмоциональное отношение пользователя к конкретному участку дороги и/или POI.

Все три перечисленные выше пункта позволяют более точно персонализировать описание маршрута, сделав ее максимально удобным для использования [6]. Благодаря им:

- при построении описания маршрута будут использоваться статистические частотные данные, позволяющие использовать наиболее знакомые пользователю пути и POI [13, 14, 15], несмотря даже на возможную меньшую оптимальность траектории движения (из предположения, что чем чаще человек пользуется одним и тем же маршрутом, тем более он хорошо ему знаком и может быть описан менее детально, нежели новый неизвестный);
- вводимое пользователем в ручном режиме эмоциональное отношение к участкам пути и/или конкретным объектам на нем дает возможность создать такой вариант описания маршрута, который будет казаться ему более коротким и оптимальным [5] для запоминания (основано на подтвержденных исследованиях в системе виртуальной CAVE, проводимое факультетом психологии МГУ им. М.В. Ломоносова).

### 2.1.5 Проблема склонений слов

Для формирования правильных с точки зрения русского языка текстов необходимо использовать склонения названий POI. То есть неправильная форма «доехать до Государственная Дума РФ» должна быть представлена в корректном виде «доехать до Государственной Думы РФ» [37].

В разделе о технической реализации сделано сравнение существующих решений и показано, что они мало подходят для склонений названий POI — в своей основе они ориентированы на преобразования существительных или фамилий, имен, отчеств. Поэтому был проведен анализ существующих лингвистических материалов и выработаны основные правила, корректные с точки зрения русского языка [42].

**Русские географические названия.** Географическое название, употребленное с родовыми наименованиями город, село, деревня, хутор, река и др., выступающее в функции приложения, согласуется с определяемым словом, то есть склоняется, если топоним русского, славянского происхождения или представляет собой давно заимствованное и освоенное наименование. Правильно: в городе Москве, в городе Санкт-Петербурге, из города Киева; в деревню Ивановку, из деревни Ольховки, в селе Шушенском, под хутором Михайловским; у реки Волги, долина ручья Сухого.

Склоняются обе части в наименовании Москва-река: Москвы-реки, на Москве-реке и т. д. В разговорной речи встречаются случаи несклоняемости первой части: за Москва-рекой, на Москва-реке и т. д. Но такое употребление не соответствует строгой литературной норме.

Географические названия в сочетании с родовым словом обычно не склоняются в следующих случаях:

- когда внешняя форма названия соответствует форме мн. числа: в городе Великие Луки, в городе Мытищи;

- когда род обобщающего нарицательного слова и топонима не совпадают: на реке Енисей, у реки Хопёр, в деревне Парфёнок (однако это замечание не относится к сочетаниям со словом город, поэтому правильно: в городе Туле, из города Москвы; о целесообразности употребления здесь самого слова город см. ниже).

Кроме того, не склоняются приложения-топонимы среднего рода, оканчивающиеся на -е, -о: между селами Молодечно и Дорожно, в городе Видное (это название не склоняется, т. к. при склонении сложно будет восстановить исходную форму: в городе Видном – это город Видный или город Видное?) [38].

Топонимы славянского происхождения, оканчивающиеся на -ово, -ево, -ино, -ыно, не склоняются в сочетании с родовым словом: из района Люблино, в сторону района Строгино, к району Митино, в городе Иваново, из деревни Простоквашино, до края Косово. Если же родового слова нет, то возможны оба варианта, склоняемый и несклоняемый: в Люблине и в Люблино, в сторону Строгина и в сторону Строгино, в Иванове и в Иваново, из Простоквашина и из Простоквашино, до Косова и до Косово, к Митину и к Митино, 8-й микрорайон Митина и 8-й микрорайон Митино. При этом склоняемый вариант соответствует строгой литературной норме. Словарь Л. К. Граудиной, В. А. Ицковича, Л. П. Катлинской «Грамматическая правильность русской речи» указывает: «В образцовом литературном стиле (со сцены, с телеэкрана, в радиоречи) эти формы следует склонять» [39].

Географические названия на -ов (-ев), -ово (-ево), -ин, -ино (-ыно) имеют в творительном падеже окончание -ом, например: Львов – Львовом, Канев – Каневом, Крюково – Крюковым, Камышин – Камышином, Марьино – Марьином, Голицыно – Голицыном.

В отличие от названий городов русские фамилии на -ин (-ын) и на -ов (-ев) имеют в творительном падеже единственного числа окончание -ым, ср.: Пушкин (фамилия) – Пушкиным и Пушкин (город) – Пушкином; Александров (фамилия) – Александровым и Александров (город) – Александровом.

Если сложносоставной топоним представляет собой русское или давно освоенное название, в косвенных падежных формах его первая часть должна склоняться: из Камня-Каширского, в Переславле-Залесском, в Могилеве-Подольском, в Ростове-на-Дону. То же в сочетании с родовым термином: в городе Петропавловске-Камчатском, в городе Ростове-на-Дону [40].

Все топонимы, у которых первая часть названия имеет морфологическую примету среднего рода, охвачены тенденцией к неизменяемости: из Ликино-Дулева, в Соболево-на-Камчатке.

**Иноязычные географические названия.** Названия, которые оканчиваются на -а:

- многие заимствованные географические названия, освоенные русским языком, склоняются по типу сущ. жен. рода на -а, например: Бухара – в Бухаре, Анкара – до Анкары;
- не склоняются французские по происхождению топонимы, оканчивающиеся на -а в языке-источнике: Гра, Спа, Ле-Дора, Юра и т. п. Однако названия, к которым в русском языке было прибавлено окончание -а, склоняются: Тулуза, Женева, Лозанна – в Тулузе, Женеве, Лозанне (ср.: Toulouse, Genève, Lausanne);
- склоняются японские географические названия, оканчивающиеся на -а безударное: Осака – в Осаке, Фукусима – из Фукусимы;
- не склоняются эстонские и финские наименования: из Ювяскюля, на Сааремаа;
- испытывают колебания при склонении абхазские и грузинские топонимы, оканчивающиеся на безударное -а. Тем не менее многие из подобных названий склоняются: Очамчира – в Очамчире, Гудаута – до Гудауты, Пицунда – из Пицунды;
- не склоняются сложные географические названия на -а безударное, заимствованные из испанского и других романских языков: в

Баия-Бланка, в Баия-Лаура, из Херес-де-ла-Фронтера, в Сантьяго-де-Куба, от Пола-де-Лена, из Сантьяго-де-Компостела;

- склоняются сложные славянские названия, являющиеся существительными при наличии словообразовательных признаков прилагательных, например: Бяла-Подляска – из Бяла-Подляски, Банска-Бистрица – до Банска-Бистрицы.

Названия, оканчивающиеся на -о и -е не склоняются в русском литературном языке: в Осло, Токио, Бордо, Мехико, Сантьяго, Кале, Гродно, Вильно, Ковно. Большую тенденцию к склоняемости имеют топонимы на -ы: в Катовицах, Фивах, Татрах, Каннах, Чебоксарах. Обычно не склоняются названия на -и: из Чили, Тбилиси, Нагасаки. Иноязычные названия, оканчивающиеся на согласный, обычно не склоняются в функции приложения: в городе Луисвилл, в городе Мобеж, в г. Ниамет, в провинции Зядинь, близ города Мэнстон. (Исключение составляют названия, давно заимствованные и освоенные русским языком: в городе Вашингтоне.) Если же подобные названия не употреблены в функции приложения, они, как правило, склоняются: в городе Мантасас, но в 70 километрах от Мантасаса, близ города Мэнстон, но близ Мэнстона. От указанной группы отступают латиноамериканские названия на -ос: в Фуэнтос.

Не склоняются сложные наименования типа Пер-Лашез, Майн-Милл, Пуэрто-Монт. Не склоняются сложносоставные наименования со второй частью -стрит, -сквер, -парк, -палас: по Элвин-стрит, на Юнион-сквер, в зале Фридрих-штадт-палас, в Энмор-парк [41].

Первая часть сложносоставных иноязычных топонимов, как правило, не склоняется: в Алма-Ате, под Буэнос-Айресом, из Йошкар-Олы. Исключение – первая часть в конструкции «топоним на реке»: во Франкфурте-на-Майне, к Шведту-на-Одере, из Стратфорда-на-Эвоне.

Если любое иноязычное сложносоставное название употреблено в функции приложения с нарицательными словами типа город, местечко, столица, порт и под., оно и во второй части оставляется в неизменяемой форме: в городе

Санта-Крус, в боливийской столице Ла-Пас(исключение составляют давно заимствованные, освоенные русским языком названия: в городе Нью-Йорке) [41].

### 2.1.6 Проблема придания «человечности» формируемому тексту

Важной задачей алгоритма является придание тексту «человечности». Под данным термином понимается такая форма текстового описания, которая была бы воспринята человеком, как составленная другим человеком [8, 16]. Для решения этой проблемы был разработан следующий механизм формирования каркаса лексем, основанный на случайных величинах и частотах использования слов и оборотов.

Все описание разбивается на блоки  $PB = \{ACTION\_TYPE, DIRECTION, RELATIVITY\_NEAR, POI\_NAME\_NEAR, DISTANCE, RELATIVITY\_TILL, POI\_NAME\_TILL\}$  (важно сразу отметить, что не все поля  $PB$  будут использованы в итоговом сформированном текстовом описании):

- *ACTION\_TYPE* — вид действия, это может быть «проходите», «проезжайте», «двигайтесь», «поверните», «развернитесь» и так далее;
- *DIRECTION* — направление движения, это может быть «прямо», «мимо», «налево», «направо» и так далее;
- *RELATIVITY\_NEAR* — относительность действия к POI, это может быть «у», «мимо», «рядом» и так далее;
- *POI\_NAME\_NEAR* — название POI, рядом с которым происходит действие, связанное с *RELATIVITY\_NEAR*;
- *DISTANCE* — расстояние, которое необходимо преодолеть, это может быть «полкилометра», «четверть километра», «пара километров», «сразу» и так далее;
- *RELATIVITY\_TILL* — относительность действия к POI, это может быть «до», «вплоть» и так далее;

- *POI\_NAME\_TILL* — название POI, рядом с которым происходит действие, связанное с *RELATIVITY\_TILL*.

Далее алгоритм разбивает *PB* на предложения, для чего используются следующие правила:

- в одном предложении может быть минимум один и максимум три блока *PB*:
  - один блок *PB* может быть только в конечном предложении и только в том случае, если к нему не может быть добавлен другой блок *PB*;
  - в остальных случаях используется два или три блока *PB*;
  - вероятность появления трех блоков *PB* определяется значением наперед заданной константы  $\Psi_1$ .
- в случаях, когда два последовательных блока *PB* содержат одинаковые значения *ACTION\_TYPE*, они могут быть сгруппированы в один; вероятность определяется наперед заданной константой  $\Psi_2$ .

Предложения, в которых повторяется одно и то же слово или словесный оборот, воспринимаются человеком как некорректные, поэтому важно избавиться от подобных комбинаций. Для этого заведены следующие константы, упомянутые выше:

- $\Psi_1$  — вероятность предложения из трех блоков *PB*;
- $\Psi_2$  — вероятность группировки двух блоков *PB* в один.

Мусорные слова используются для придания тексту большей «человечности». Под мусорными словами понимаются наречия, используемые людьми при описании пути, например, «затем», «после чего», «далее» и другие.

Для каждой лексемы определена таблица с возможными вероятностями ее использования. Помимо этого, четко задано правило, согласно которому не может быть использовано две одинаковые словесные формы лексемы одного типа. В случаях, когда это неразрешимо, например, при необходимости указать

одинаковое расстояние на последовательных интервалах пути, используются лексем-заменители. Для каждого типа лексем, где это возможно, определены свои лексем-заменители:

- для *ACTION\_TYPE*: «снова», «и опять», «еще раз» и другие;
- для *DIRECTION*: «в ту же сторону» и другие;
- для *DISTANCE*: «столько же», «такое же расстояние» и другие.

## **2.2 Распознавание текстового описания маршрута**

Второй задачей алгоритма является распознавание текстового описания маршрута. Решение представляет собой, по сути, обратную последовательность действий с несколькими обобщениями [4], позволяющими сократить объем вводимой пользователем информации.

### **2.2.1 Описание алгоритма**

Алгоритм состоит из следующих шагов:

1. получение пользовательского ввода, задающего начальную и конечную точки маршрута:
  - а. пользователь вводит только названия POI (улицы, организации, объекты и т.д.), система в автоматическом режиме ищет подходящие и предлагает варианты во всплывающем окне или другом удобном виде;
2. получение текстового описания пути, состоящего из навигационных директив, названий POI, расстояний и даже мусорных слов:
  - а. пользователь вводит все, что ему известно о пути, при этом описание пути не обязательно должно быть полным и однозначным;
  - б. при вводе каждого нового слова система автоматически подбирает подходящие из известных ей лексем и базы

названий POI и предлагает пользователю для выбора, тем самым сокращая вероятность ошибки и ускоряя набор

3. разбиение текстового описания на блоки четверок  $PB^i = \{ACTION\_TYPE, DIRECTION, POI\_NAME, DISTANCE\}$ , которые могут быть не полностью заполнены, где:
  - a. *ACTION\_TYPE* — вид действия, которое может принимать значения вида «пройти» или «проехать», что будет обозначать вид используемого транспорта (пешком или на автомобиле), что в свою очередь повлияет на возможные варианты прокладывания пути;
  - b. *DIRECTION* — направление движения, которое может быть описано в основном только словами вида «налево» и «направо»;
  - c. *POI\_NAME* — собственно название POI, рядом с которым происходит действие;
  - d. *DISTANCE* — расстояние в приблизительном виде (очевидно, что пользователь может ошибаться), представляется обычно словами «километр», «четверть километра», «сразу» и другими;
4. прокладывание возможных вариантов маршрутов между каждой парой POI и нахождение тех интервалов, которые подходят под оставшиеся заполненные поля четверок  $PB^i$ ;
5. выставление оценки соответствия  $\Lambda_1$  пользовательскому вводу для каждой четверки  $PB^i$  и отбрасывание в случае, если она меньше  $\Theta_1$ ;
6. объединения найденных интервалов в маршруты с выставлением оценки соответствия  $\Lambda_2$  пользовательскому вводу;
7. представление всех найденных маршрутов с суммарной оценкой  $\Lambda_2$ , большей некоторого наперед заданного порога  $\Theta_2$ .

### 2.2.2 Проблема неточности введенных пользователем расстояний

Пользователь, как любой человек, может ошибаться и вводить неточную информацию. Для работы алгоритма предполагается, что пользователь точно вводит только названия POI (благодаря автоматическому дополнению текста, которое исключает ввод произвольного несуществующего названия POI), причем последовательность их упоминания правильная с точки зрения их появления на маршруте [4].

Для каждой четверки  $PB'$  между двумя заданными POI и для каждого найденного между ними маршрута:

- выставляется оценка соответствия  $\Lambda_1$  для каждой четверки  $PB'$  по правилам:
  - любое незаполненное поле четверки  $PB'$  имеет 0 баллов;
  - любое заполненное поле и являющееся частью пути получает 1 балл;
  - любое заполненное поле, но не являющееся частью пути (иными словами ошибочное), получает -1 балл.
- проверяется, что полученная оценка соответствия  $\Lambda_1$  больше наперед заданного значения  $\Theta_1$ , в противном случае происходит отбрасывание найденного интервала пути;
- все интервалы, набравшие удовлетворяющую критерий оценку, объединяются в соответствующие маршруты, целиком заполняющие путь от начальной до конечной точки;
- если такие маршруты удалось получить, то им выставляется оценка соответствия  $\Lambda_2$  по правилам:
  - оценки всех интервалов суммируются;
  - результат делится на количество интервалов, тем самым происходит нормализация оценок;

- все маршруты, получившие оценку соответствия  $\Lambda_2$  выше наперед заданного значения  $\Theta_2$ , считаются валидными и представляются пользователю для выбора.

Пользователь может вводить расстояния неточно, ошибаясь даже на порядок. Предполагается, что фактически можно найти маршрут между любыми двумя заданными точками старта и конца пути. Для решения проблемы с расстояниями применяется следующая последовательность действий:

- если ни один из найденных маршрутов не получил оценку соответствия  $\Lambda_2$  выше  $\Theta_2$ , то считается, что были неверно указаны расстояния;
- производится повторный анализ введенных четверок  $PB'$  без учета поля *DISTANCE*, найденные варианты, если они существуют, представляются пользователю с пометкой, что расстояния не были учтены.

При сравнении найденного интервала с четверкой  $PB'$  по полю *DISTANCE* происходит проверка соответствия расстояния по наперед заданному критерию возможного отклонения  $\Delta$ , указанного в процентном соотношении: все расстояния, которые отличаются от фактических на процент, больший чем  $\Delta$ , являются неподходящими.

### **2.2.3 Проблема нескольких подходящих под описание маршрутов**

Исходя из описанного выше алгоритма, возможны ситуации, когда несколько маршрутов одновременно подходят под описание. В этом случае выполняется сортировка найденных маршрутов по оценкам соответствия  $\Lambda_2$  от большего к меньшему и вывод ранжированного результата пользователю [3, 4].

## **2.3 Выводы ко второй главе**

Во второй главе рассмотрены разработанные автором алгоритмы, позволяющие строить и распознавать текстовое описание маршрута,

обладающего персонализацией, близостью к прямой речи человека и лаконичностью, стремящиеся максимально задействовать когнитивные процессы пользователя с целью упрощения восприятия информации. Благодаря текстовому формату представления они могут быть реализованы и использованы на максимально широком списке различных платформ, начиная от мобильных и стационарных компьютеров и заканчивая антропоморфными роботами. Текст всегда может быть синтезирован в голосовую речь, распечатан или выведен на экран. Появление персональных ассистентов, таких как «Siri» [43] и «OK Google» [44], подталкивает к использованию именно голосового взаимодействия с устройствами. В следующей главе рассмотрена реализация предложенных алгоритмов в виде программного модуля.

## Глава 3. Программная реализация

В третьей главе рассмотрена архитектура и реализация программного обеспечения, имплементирующего описанные выше алгоритмы. Вначале формируются общие требования к компьютерной системе. Далее на их основе прорабатывается архитектура программного решения. В главе также рассмотрены проблемы, которые возникали во время реализации алгоритмов в коде.

### 3.1 Требования к программной реализации

Перед началом выбора технологий и проведения их сравнительного анализа необходимо выработать общие требования к финальному виду программной системы, которые могут кардинально повлиять на рассматриваемые технологии [45]. Для этого были выбраны самые важные, по мнению автора, аспекты:

- кроссплатформенность — возможность запускать приложение на широком спектре устройств [49];
- платформонезависимый программный код — сокращение трудозатрат по поддержке новых платформ за счет использования существующих технологий по абстрагированию от конечных сред исполнения программ [49];
- клиент-серверная реализация — дистрибуция актуального программного кода при каждом запуске приложения, отказ от долгих обновлений и выводов новых версий приложения [48];
- повторное использование существующего банка программных API — сокращение трудозатрат по пересозданию уже написанных, отлаженных и зарекомендовавших себя систем; некоторые части ПО бывает проще адаптировать под существующую библиотеку, чем писать собственную библиотеку, приспособленную под приложение [46, 47].

Далее рассмотрим более конкретно каждый из пунктов и проведем сравнительный анализ существующих технологий, удовлетворяющий определенным выше требованиям.

**Кроссплатформенность.** В современной сети устройств присутствуют как мобильные, так и стационарные вычислительные системы. Первым требованием к архитектуре является максимальная поддержка всех систем при наименьших дополнительных затратах на разработку под конкретную платформу. Данный пункт опирается на технологии по созданию платформонезависимого программного кода и не содержит анализов технологий.

**Платформонезависимый программный код.** Исходя из предыдущего требования, необходимо выбрать способ достижения кроссплатформенности [49]. Это достижимо несколькими способами:

- *платформонезависимый интерпретируемый программный код* (исходный код программы исполняется с помощью специальной программы-интерпретатора по мере чтения файла или поступления новых команд на стандартный поток входа):
  - работает на любой платформе, обладающей нужным интерпретатором определенного языка;
  - не требует компиляции и трансляции;
  - позволяет осуществлять быструю итеративную разработку с перманентной подменой кода во время работы;
  - популярные поддерживаемые языки для написания кода: JavaScript, PHP, Python.
- *платформонезависимый компилируемый программный код* (исходный код программы преобразуется компилятором в промежуточное представление (байт-код, intermediate language) или машинный код и

записывается в файл для последующего исполнения этого файла виртуальной машиной или операционной системой):

- работает на любой платформе, под которую был скомпилирован;
- требует обязательную компиляцию;
- не позволяет осуществлять быструю итеративную разработку;
- популярные поддерживаемые языки для написания кода: Java, C#, C, C++.

Разработка программного кода велась только автором диссертации, поэтому были актуальны следующие критерии:

- минимизация временных затрат по поддержке дополнительных платформ;
- минимизация временных затрат на компиляцию, трансляцию и отладку кода;

При этом критерий производительности программного кода с учетом огромных вычислительных мощностей современных мобильных и стационарных систем был наименее существенным. Таким образом, под выбор попали следующие технологии с привязанными к ним языками программирования:

- Java — приложения компилируются в так называемый байт-код, который исполняется специальной программой JVM (Java Virtual Machine), реализации которой существуют практически для любой операционной системы и архитектуры процессора [50];
- C# — приложения компилируются аналогично Java в специальное представление, именуемое IL (Intermediate Language), реализаций виртуальных машин для исполнения IL значительно меньше, чем JVM [51];
- JavaScript — приложения не компилируются, а интерпретируются по ходу выполнения, что дает преимущество в скорости разработки; для

исполнения требуется js-движок, который интегрирован во все современные веб-браузеры, доступные практически для любых платформ, операционных систем и архитектур процессоров [52];

- PHP — приложения, аналогично JavaScript, интерпретируются на ходу при помощи специального php-движка, технология используется преимущественно для создания серверной части приложений и не имеет широко распространенных возможностей для запуска на клиентских компьютерах [53];
- Python — приложения, аналогично JavaScript и PHP, интерпретируются по ходу выполнения, могут запускаться под всеми широко распространенными платформами и архитектурами, однако в большинстве случаев (исключение составляют операционные системы семейства Linux) компьютеры не имеют предустановленного python-интерпретатора, а также на мобильные системы семейства iOS их установка запрещена правилами разработчика [54].

**Клиент-серверная реализация.** С целью упрощения передачи обновленных исполняемых файлов на разные платформы необходимо консолидировать их на одном устройстве — *сервере*, передающем их через сеть Интернет *клиентам* (устройствам, непосредственно за которым работает пользователь). Такой подход позволяет значительно сократить время разработки одной итерации и последующих обновлений исходного кода. В качестве клиентов подразумевается использование широкого набора различных платформ с индивидуальными архитектурами и отличными операционными системами. Рассмотрим с учетом данного требования перечисленные выше языки и используемые совместно с ними технологии:

- Java — программы, написанные с использованием данной технологии, потребуют перекомпиляции для работы на мобильных устройствах семейства Android, для платформ с операционной

системой iOS потребуется использовать дополнительные программные прослойки, позволяющие запустить приложение; помимо этого, повторная компиляция для разных платформ значительно увеличит время одной итерации разработки, либо сузит список поддерживаемых устройств [50];

- С# — программы, написанные с использованием данной технологии, не поддерживаются ни одной современной распространенной мобильной операционной системой, для их запуска требуется использование специальных промежуточных библиотек [51];
- JavaScript — программы, написанные с использованием данной технологии, могут быть запущены на любой популярной и распространенной платформе внутри веб-браузера; для этого языка не требуется компиляция и возможна быстрая итерация путем запроса нового кода приложения от сервера на каждом запуске программы [52];
- PHP — программы, написанные с использованием данной технологии, не могут быть запущены без дополнительных библиотек на разных мобильных системах и будут требовать использования платформозависимых API [53];
- Python — аналогично PHP, программы не могут быть запущены на разных мобильных системах [54].

### **Повторное использование существующего банка программных API.**

Помимо широкого спектра библиотек, доступных для свободного использования, в наши дни стали популярны онлайн сервисы, предоставляющие разнородные бесплатные услуги, упрощающие создание приложений [46]. К таким относятся:

- картографическо-навигационные системы, выдающие данные по маршрутам, POI и т.д.;

- облачные хранилища данных, позволяющие получить доступ к сохраненной информации с любого устройства;
- социальные сети, позволяющие получить список контактов и историю перемещений между POI;
- другие.

Были рассмотрены способы взаимодействия с самыми известными системами, которые могли потребоваться для реализации алгоритмов:

- Google Maps — навигационно-картографический сервис [33];
- Bing Maps — навигационно-картографический сервис [35];
- Яндекс Карты — навигационно-картографический сервис [34];
- Карты Mail.Ru — навигационно-картографический сервис [55];
- Dropbox — облачное файловое хранилище [56];
- Vox — облачное файловое хранилище [57];
- Google Drive — облачное файловое хранилище [74];
- Облако Mail.Ru — облачное файловое хранилище [58];
- Foursquare — социальная сеть [59];
- Facebook — социальная сеть [60];
- ВКонтакте — социальная сеть [61].

Все рассмотренные системы поддерживают систему REST [62] запросов по протоколу HTTP/HTTPS в формате JSONP. Все из рассматриваемых выше языков программирования поддерживают работу с данным форматом запросов.

**Выбранные технологии.** По итогам рассмотрения технологий для реализации был выбран язык программирования JavaScript в его платформонезависимом интерпретируемом представлении в сочетании с технологиями HTML5 и языком разметки пользовательского интерфейса HTML в связке с CSS [63]. На серверной стороне использован сервер Apache. На клиентской стороне роль интерпретатора выполняет веб-браузер.

Пользовательские данные хранятся как локально в системе Storage веб-браузера, так и в облачном хранилище. Доступ к API онлайн-сервисов осуществляется по протоколу REST в формате JSONP по технологии Ajax. Используется поддержка безопасного зашифрованного протокола передачи данных HTTPS с возможностью возврата к HTTP.

### **3.2 Архитектура программного решения**

По итогам рассмотрения технологий в предыдущем параграфе была разработана архитектура, состоящая из следующих компонент:

- сервер — предназначен для хранения исходных кодов приложения с реализацией алгоритмов на языке JavaScript и интерфейсом на технологии HTML5, передаваемых на клиентские устройства для исполнения, при этом сервер никаких вычислений не производит;
- клиент(ы) — мобильные или стационарные компьютеры, имеющие установленный веб-браузер преимущественно последней версии (для полной совместимости и большей вероятности поддержания стандарта HTML5, принятого в текущем году, например Firefox, Chrome, Opera, Safari), исполняют полученный от сервера код в автономном от сервера режиме, т.е. передача данных от сервера к клиенту осуществляется только один раз в момент загрузки приложения, все остальные данные получают от онлайн-сервисов путем осуществления дополнительных запросов
- сервисы — доступные бесплатные онлайн-сервисы, такие как: навигационно-картографические системы, облачные хранилища файлов, социальные сети.

Передача данных осуществляется по протоколам HTTP и HTTPS. HTTP, не предоставляющий средств шифрования [64], используется для передачи исходных кодов программы на клиент с целью уменьшения нагрузки на сервер. HTTPS,

предоставляющий шифрование, используется для передачи личной информации из социальных сетей и для обмена файлами с облачным хранилищем.

Потоки передачи данных приложения сконцентрированы вокруг клиентской стороны и имеют ориентировочно следующую последовательность (рис. 24):

1. Из веб-браузера клиента происходит запрос на сервер по определенному URL адресу по протоколу HTTP методом GET;
2. Сервер обрабатывает запрос и отправляет клиенту index.html файл, содержащий в себе разметку пользовательского интерфейса, ссылки на файлы с исходными кодами на языке JavaScript и визуальными стилями оформления на языке CSS3;
3. Клиент получает index.html файл, делает дополнительные запросы на получение файлов, указанных в нем в виде ссылок, аналогично по протоколу HTTP методом GET;
4. Сервер отправляет запрошенные файлы, на этом общение сервера и клиента заканчивается;
5. Клиент отправляет запрос в навигационно-картографическую систему с данными по пути, необходимыми для работы алгоритмов, по протоколу HTTPS методом GET, используя технологию REST запросов;
6. Сервер навигационно-картографической системы отвечает, передавая необходимые данные клиенту;
7. Возможны дополнительные запросы в социальные сети и облачные файловые хранилища аналогично пунктам 5 и 6.



Рис. 24: Схема архитектуры программного решения

### 3.3 Критерии выбора веб-фреймворков

Для решения проблем, возникающих при реализации, разумно воспользоваться существующими фреймворками, главной задачей которых является упрощение и ускорение разработки. Для грамотного выбора подходящего программного решения были выработаны критерии:

- широкая поддержка веб-браузеров;
  - позволяет работать однажды написанному коду на максимально большом списке веб-браузеров, как поддерживающих стандарт HTML5, так и нет; обязательными для поддержки были Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome (параметр версии веб-браузера на данной стадии разработки не является ключевым, предполагается, что пользователь использует последние версии программного обеспечения);
- поддержка мобильных устройств;

- позволяет работать однажды написанному коду как на стационарных компьютерах, так и на мобильных без заметных отличий;
- открытость исходных кодов;
  - позволяет исследовать технологию изнутри и в случае необходимости дописать требующийся функционал;
- свободная лицензия;
  - позволяет вести бесплатную некоммерческую и коммерческую разработку;
- наличие документации;
  - позволяет сконцентрироваться на разработке собственных исходных кодов и использовать технологию как «черный ящик», предлагающий функционал, описанный в документации вместо исследования его исходных кодов.

Описанные в данном параграфе критерии используются далее по тексту при сравнительных обзорах существующих технологий.

### **3.4 Проблема повторного использования функционального кода**

Стандарт HTML5, включающий в себя язык программирования JavaScript, язык разметки интерфейса HTML и язык визуальных стилей CSS, был принят в начале 2015 года и на текущий момент не полностью поддерживается разными версиями веб-браузеров. При этом реализация существующей поддержки немного отличается в каждом веб-браузере. Для повторного использования функционального кода был проведен анализ существующих решений данной проблемы.

Были рассмотрены фреймворки:

- jQuery [65]

- широкая поддержка веб-браузеров: да
- поддержка мобильных платформ: да
- открытость исходных кодов: да
- свободная лицензия: да
- наличие документации: да
- KendoUI [66]
  - широкая поддержка веб-браузеров: да
  - поддержка мобильных платформ: да
  - открытость исходных кодов: да
  - свободная лицензия: нет
  - наличие документации: да
- AngularJS [67]
  - широкая поддержка веб-браузеров: да
  - поддержка мобильных платформ: да
  - открытость исходных кодов: да
  - свободная лицензия: да
  - наличие документации: да

По итогам сравнения был выбран фреймворк jQuery, как наиболее широко удовлетворяющий поставленным критериям.

### **3.5 Проблема адаптивного интерфейса пользователя**

Экраны мобильных и стационарных компьютеров имеют разные размеры. Методом ввода на стационарных компьютерах обычно является манипулятор «мышь», на мобильных платформах преимущественно используются тач-дисплеи, и ввод осуществляется непосредственно касанием нужной области экрана. Обе отличительные особенности создают проблемы при создании пользовательского интерфейса. Как и в остальных ситуациях стоял вопрос о создании кода UI только один раз, чтобы он автоматически адаптировался под платформу (рис. 25). Для достижения данной цели были исследованы существующие фреймворки,

нацеленные на решение данной проблемы. Все фреймворки удовлетворяли выбранным критериям сравнения и имели лишь незначительные отличия. Поэтому было принято решение учесть мнение пользователей и сравнить их отзывы на сайте GitHub [73], где хранятся исходные коды фреймворков.

Были рассмотрены фреймворки:

- Bootstrap (75000+ звезд) [68]
- Foundation (18000+ звезд) [69]
- Semantic (12900+ звезд) [70]
- Pure (9900+ звезд) [71]
- UIKit (3800+ звезд) [72]

По итогам сравнения был выбран явный лидер — Bootstrap, предпочтение которому отдает большинство разработчиков.

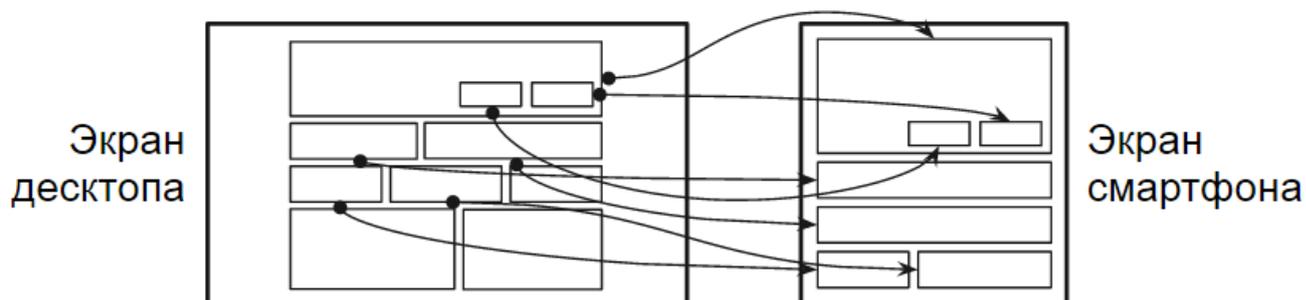


Рис. 25: Схема изменения адаптивного пользовательского интерфейса под размеры экрана

### 3.6 Проблема сохранения пользовательских данных

К пользовательским данным в рамках созданного приложения относятся:

- данные по перемещениям — массив двоек {географические координаты; отметка времени};
- данные по посещенным местам — массив троек {географические координаты; название места; отметка времени};

- данные по знакомым маршрутам — массив доек {массив географических координат, описывающих траекторию пути; выбранное пользователем название}

Данные пользователя необходимо хранить как локально на устройстве, так и в некотором облачном хранилище для доступа к ним с любого устройства. Для грамотного выбора облачного хранилища был сделан их сравнительный анализ по следующим критериям:

- наличие бесплатного пространства для хранения данных;
- наличие программного REST API или JavaScript библиотеки для доступа к данным без использования собственной серверной стороны;

Сравнению подлежали следующие онлайн сервисы:

- Dropbox [56]
  - наличие бесплатного пространства для хранения данных: да;
  - наличие программного REST API или JavaScript библиотеки: да;
- Box [57]
  - наличие бесплатного пространства для хранения данных: да;
  - наличие программного REST API или JavaScript библиотеки: нет;
- Google Drive [74]
  - наличие бесплатного пространства для хранения данных: да;
  - наличие программного REST API или JavaScript библиотеки: да;
- Облако Mail.Ru [58]
  - наличие бесплатного пространства для хранения данных: да;
  - наличие программного REST API или JavaScript библиотеки: нет;

По итогам сравнения был выбран Dropbox, как обладающий большей распространенностью по сравнению с Google Drive, однако система поддерживает легкий переход на Google Drive.

### **3.7 Функциональные возможности**

В итоговой программной реализации имеется два функциональных компонента: построение текстового описания (рис. 27) и распознавание текстового описания (рис. 28). Примеры снимков экрана работающей программы можно найти в Приложении 1. Для построения текстового описания используется интерфейс пользователя со следующими задаваемыми полями:

- начальная точка пути;
- конечная точка пути.

Оба поля имеют автоматические подсказки, то есть пользователь начинает вводить название POI и сразу получает список подходящих по названию объектов (рис. 26).

Далее, после подтверждения ввода, в действие вступает алгоритм. Он получает необходимую информацию у онлайн-навигационно-картографической системы и строит текстовое представление в формате HTML. На рис. 29 нарисован пример пути, представленный в виде изображения траектории с метками, используемыми для его перевода в текстовое представление.

Благодаря использованию HTML формата для представления текстового описания возможно (рис. 30):

- создание выделенных жирным и курсивным шрифтом фрагментов текста (такой стиль оформления имеют знакомые пользователю POI или маршруты);
- создание всплывающих окон с дополнительной информацией по POI (формальное название, описание внешнего вида и изображение с той

точки, с которой будет видно POI пользователю в процессе перемещения по маршруту) при наведении курсора мыши на активный элемент или нажатия на него пальцем.

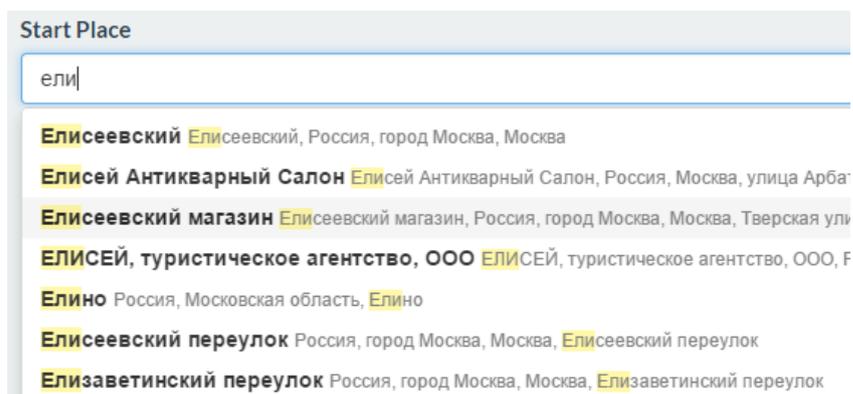
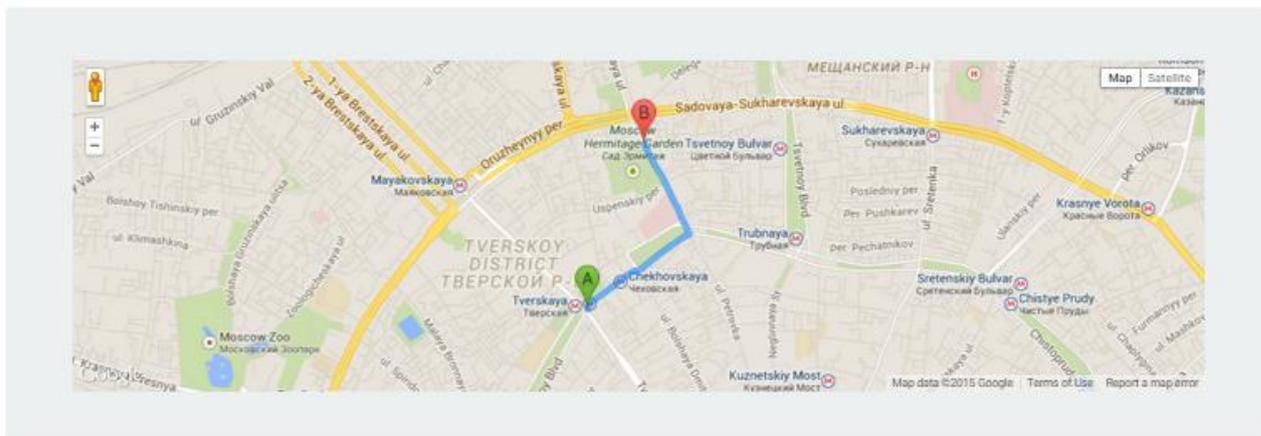


Рис. 26: Автоматическое дополнение названий POI

Start Place  
 Елизеевский магазин, Россия, город Москва, Москва, Тверская улица

End Place  
 Новая Опера, Россия, Москва, улица Каретный Ряд

Draw POIs Show Directions



**Directions by Text**

Проезжайте полкилометра прямо по Страстному бульвару мимо Театрального центра, поверните налево у ресторана Ти-Бон, затем снова полкилометра прямо, вы на месте.

**Directions by Google**

Walking directions are in beta. Use caution – This route may be missing sidewalks or pedestrian paths.

A Strastnoy bulvar, 4 строение 4, Moskva, Russia, 125009

1.1 km - about 14 mins

1. Head northeast on бул. Страстной toward Страстной б-р	32 m
2. Continue onto Страстной б-р	11 m
3. Continue onto бул. Страстной	0.2 km
4. Slight right onto Страстной б-р	0.3 km
5. Turn left onto ул. Петровка	0.3 km
6. Continue onto ул. Каретный Ряд	0.2 km

B ulitsa Karetnyy Ryad, 3, Moskva, Russia, 127006

Map data ©2015 Google

Рис. 27: Интерфейс пользователя для задания маршрута и получения текстового описания

## Path Description

**Start Place**  
 Елисейевский магазин, Россия, город Москва, Москва, Тверская улица

**End Place**  
 Новая Опера, Россия, Москва, улица Каретный Ряд

**Directions**  
 прямо полкилометра до Place 1 затем налево полкилометра

**Place 1**  
 Театральный Центр Стд РФ, Россия, город Москва, Москва, Страстной бульвар

**Place 2**  
 Ти-Бон, Россия, Москва

**Place 3**  
 Pick a place location...

Рис. 28: Интерфейс пользователя для ввода текстового описания маршрута

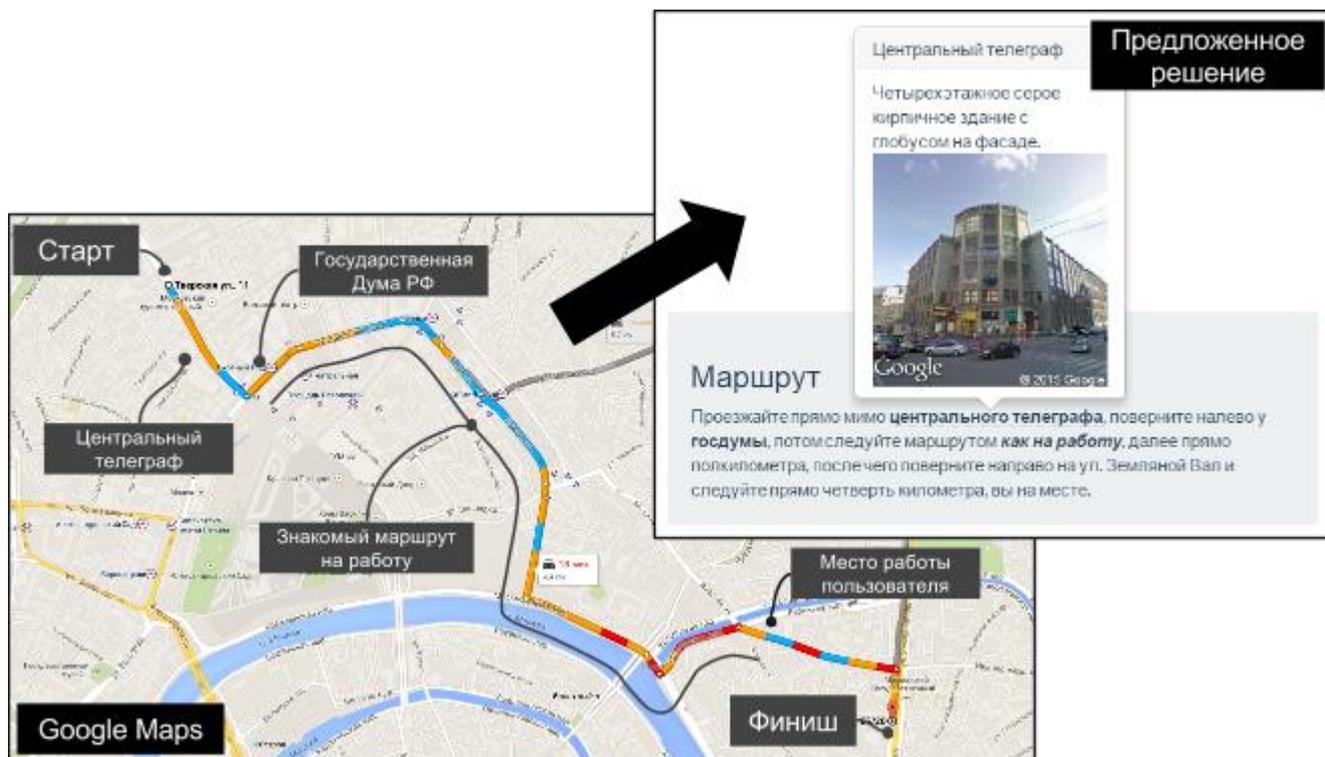


Рис. 29: Пример перевода изображения траектории пути в текстовое представление с использованием знакомых пользователю маршрутов и POI



Рис. 30: Составные части текстового описания пути, используемые алгоритмом

Для распознавания текстового описания используются следующие поля:

- начальная точка пути;
- конечная точка пути;
- описание маршрута;
- список POI, используемых в описании.

Пользователь задает начальную и конечную точку маршрута, далее начинает описывать маршрут текстом в специальном поле. При осуществлении ввода всплывают подсказки с вариантами слов, которые он может использовать с целью сокращения числа ошибок (рис. 31). В самом описании POI используются не напрямую, а через ссылки на поля ниже, в которых указываются точные названия POI с помощью автоматических подсказок (аналогично точкам начала и окончания пути).

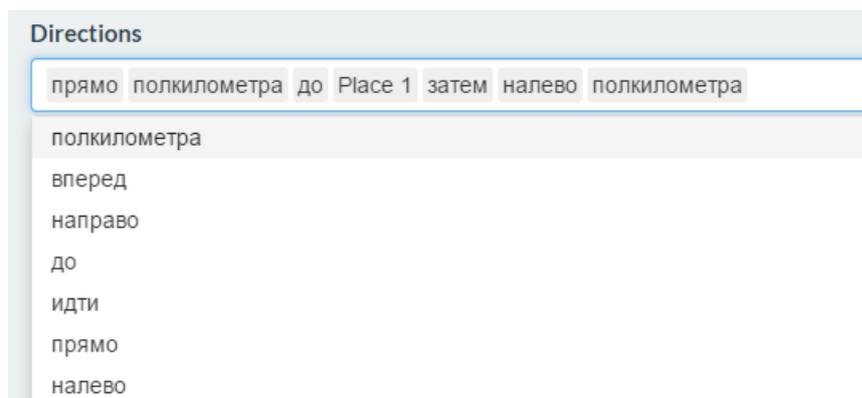


Рис. 31: Автоматическое дополнение при описании маршрута

### 3.8 Практическое применение

Разработанная программная реализация выполняет демонстрационную задачу и показывает пригодность и удобство описываемого в диссертации метода построения и распознавания текстового описания маршрута.

Описанные выше технологии задействованы в разработке программы управления роботом РБ-2 (рис. 32). Робот предназначен для использования как в

очевидных задач, таких как робот-гид, патрульный робот, робот телеприсутствия, робот для проведения исследований в области робототехники, робот для обучения школьников и студентов и т.п., так и в таких, где нахождение человека нежелательно или невозможно. Например:

- контроль/сервис на автоматических станциях в удаленных, труднодоступных местах;
- удаленная помощь больным, престарелым.



Рис. 32: Многофункциональный робот РБ-2

Робот РБ-2 представляет собой платформу с манипулятором и умеет выполнять следующие задачи (не все пункты полностью завершены, робот находится в стадии разработки):

- функционировать полуавтономно, управляться как простыми речевыми командами, так и удаленно, по общедоступному каналу связи, например, интернету;

- обладать многоуровневой базой данных с информацией об окружающих средах, в которых он будет функционировать;
- обладать многоуровневой базой данных с навыками простых действий и манипуляций, которые могут выполняться им самостоятельно, но инициироваться командами человека-оператора;
- осуществлять навигацию по заранее автоматически построенной (на основе пассфреймов [79]) и размеченной карте местности (рис. 33) и перемещаться по строго заданной траектории (при этом траектория задается голосом человеком-оператором в свободной форме).

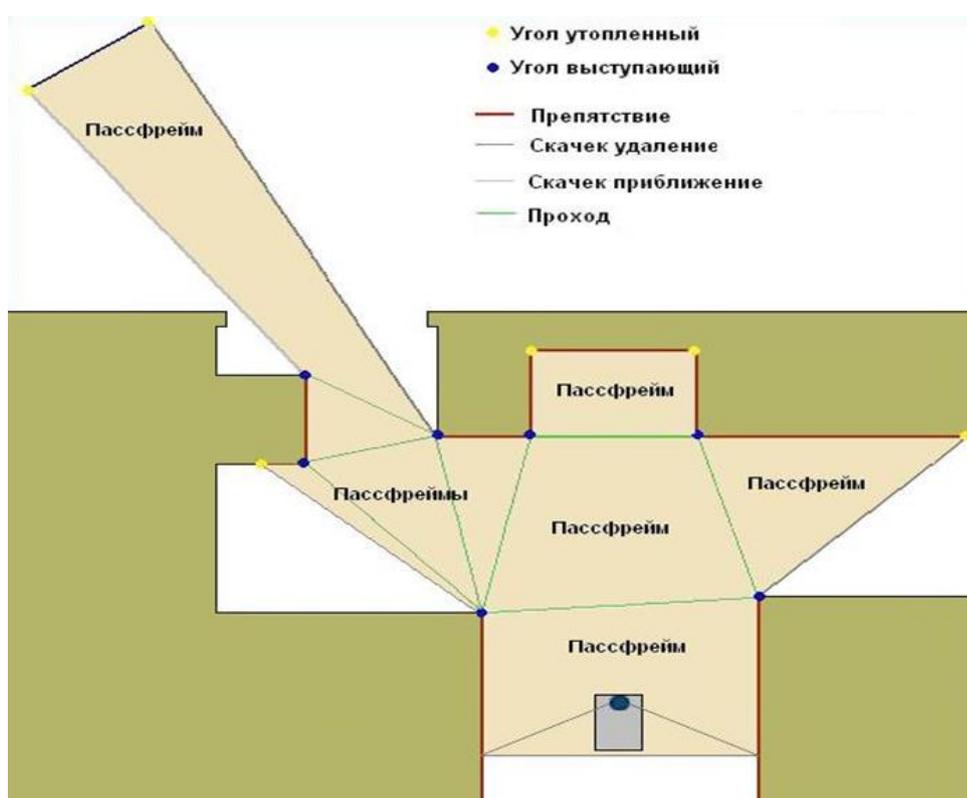


Рис. 33: Пример построенной роботом карты окружения с использованием технологии пассфреймов

Аналогичные, но немного упрощенные, версии алгоритмов были использованы при построении летающего дрона (квадрокоптера), участвовавшего в конкурсе роботов, проводимого компанией КРОК (рис. 34). Устройство

перемещалось по заранее заданному простому лабиринту, однако правила его перемещения были заданы в близком к естественному языку виде. Благодаря этому появилась возможность быстро адаптировать алгоритм поведения устройства прямо на площадке перед началом соревнований.

Робот обладал сенсорами расстояний, расположенными в пяти позициях (по четыре в плоскости полета, направленные в разные стороны, и один, смотрящий строго вниз), и одной камерой, направленной вперед и вниз. При помощи датчиков расстояния и заранее известной приблизительной структуры уровня робот мог позиционировать себя в пространстве. Далее в ход вступал описанный в предыдущих частях диссертации метод распознавания описания маршрута, но с одним отличием. Вместо перебора подходящих под описание путей робот начинал сразу следовать по первому подходящему и принимал решения о выборе следующего шага в режиме реального времени. Таким образом, благодаря введенному описанию маршрута робот мог развивать большую скорость в тех местах, где не было указано никаких препятствий, и, наоборот, замедляться и аккуратнее перемещаться в областях с перегородками уровня.



Рис. 34: Автономный квадрокоптер, участвовавший в конкурсе летающих роботов от компании КРОК

Технологии, задействованные при разработке программного модуля, используются также для создания расширения навигационно-картографической онлайн-системы Карты Mail.Ru (рис. 35). Подробные детали не могут быть описаны по причине конфиденциальности корпоративной информации.

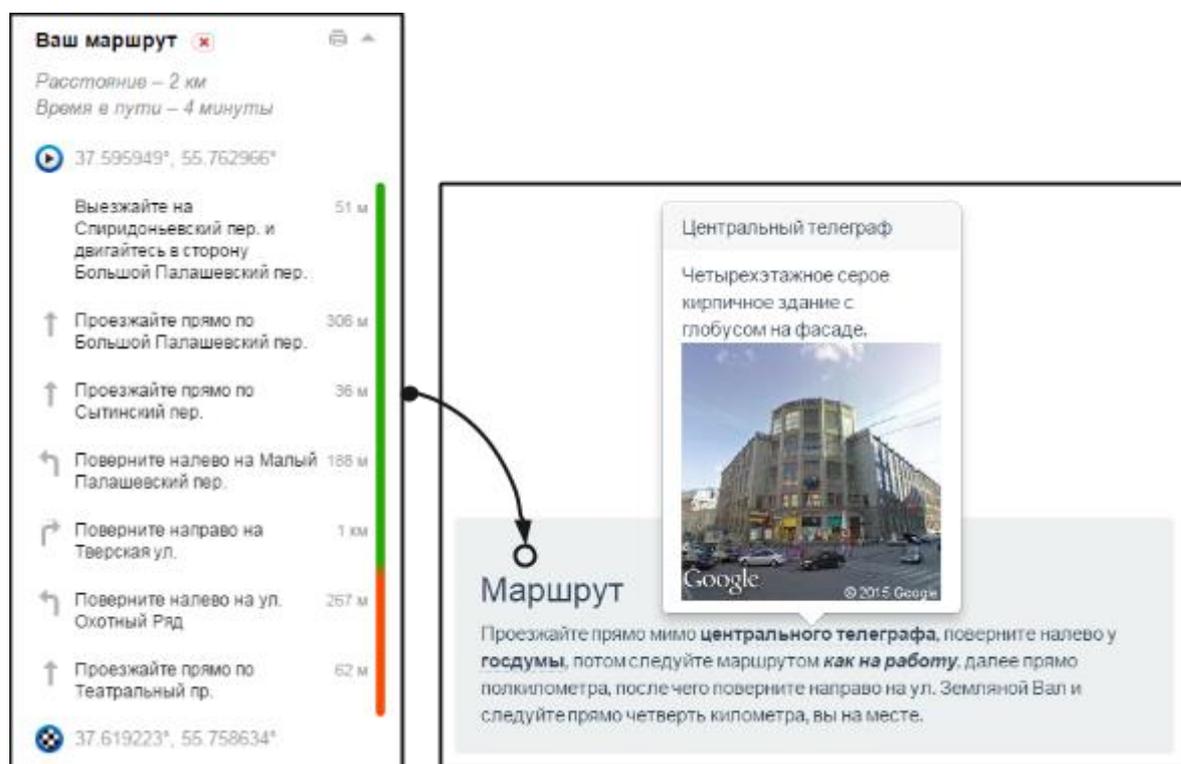


Рис. 35: Пример перевода последовательности действий к перемещению, генерируемых навигационно-картографической системой Карты Mail.Ru в ее текстовое описание

Ответвление реализованного приложения, разработанное автором диссертации, было использовано для навигации внутри офисного здания Skylight (рис. 36) [3]. Особенностью данного здания является полная симметрия по четырем направлениям и отсутствие каких-либо ориентиров и указателей на этажах. Так как техническое оснащение здания разного рода табличками и надписями требует времени и внимательной проработки деталей, было принято решение создать компьютерную систему, решающую данную проблему в большей части случаев. Система работает через локальную сеть Интранет, не

доступную из внешней среды. Доступ к ней осуществляется через корпоративный портал компании Mail.Ru, на котором у любого сотрудника есть персональный профиль с открытой для других работников информацией: фотография, занимаемая должность, адрес рабочего места и другая информация. Рядом с указанием адреса была добавлена кнопка, показывающая текстовое описание пути к нему, исходя из предположения, что пользователь уже добрался до нужного этажа на лифте (лифты располагаются в центре здания и путь до них не имеет смысл описывать, он хорошо знаком каждому, кто находится внутри офиса). Для построения текстового описания пути к рабочему месту сотрудника был проведен опрос на предмет объектов, которые могли бы стать ориентирами на этажах. В качестве подобных объектов были выбраны две сущности:

- автосалон компании Mercedes, на здании которого расположен хорошо заметный из окон офиса вращающийся логотип компании;
- направление в сторону области, так как из окон хорошо видно Ленинградское шоссе;

В офисе имеется 6 лифтов, все они располагаются в одном холле и имеют номера от 1 до 6, при этом 1-3 расположены по одной стороне, 4-6 — по другой. Для вызова лифтов используются специальные терминалы, расположенные рядом с ними — 2 штуки на этаж. Экран терминала чувствителен к нажатиям пальцами. В качестве изображения выводится список всех этажей здания — 26 позиций. В случае, если сотрудник желает поехать на определенный этаж, он должен нажать на него на терминале и тем самым отправить свою заявку в систему управления лифтами. В ответ на его запрос на экране терминала будет отображен номер лифта, который довезет его до нужного этажа. Внутри лифта нет способа выбрать желаемый этаж. Благодаря такому подходу возможно программное управление сразу всеми лифтами здания — компьютерная система знает, где находится какой лифт, сколько приблизительно в нем людей и на какие этажи он едет, поэтому предоставит пользователю лифт в кратчайшие сроки.

В системе построения описания пути к рабочему месту используется статическое представление текста — то есть для разных пользователей будет показан один и тот же текст. Это связано с двумя факторами:

- описание маршрута ведется от лифта на этаже, где работает искомый сотрудник, к которому пользователь желает пойти;
- заранее невозможно предугадать, какой лифт будет предоставлен для поездки.



Рис. 36: Пример текстового описания пути к рабочему месту сотрудника с началом от лифтового холла внутри офисного здания Skylight

Ввиду этого, в текстовом описании за основу предполагается, что пользователь всегда перемещается по этажам на лифтах 1-3. В качестве уточнения, в какую сторону выходить из лифта, используется выбранный ориентир. Логично, что в случае, если пользователю будут выдан лифт 4-6, то первое действие по выходу из лифта будет необходимо инвертировать (вместо «из лифта налево» будет «из лифта направо»).

### 3.9 Сравнение

Ввиду специфики данного исследования и отсутствия схожих систем построения и распознавания текстового представления маршрута в качестве методики сравнения был выбран опрос пользователей при помощи анкетирования. Сравнению подлежали:

- методы построения описания маршрута:
  - предложенный алгоритм построения текстового описания;
  - изображение траектории;
  - пошаговая инструкция к перемещению.
- методы распознавания описания маршрута:
  - предложенный алгоритм распознавания текстового описания;
  - ввод траектории через опорные точки, доступный на стационарных компьютерах.

Ввиду специфики предмета опроса (удобства использования) была дополнительно задействована «Модель Кано» [77, 78] — метод, разработанный Нориаки Кано, используемый для оценки эмоциональной реакции пользователя на отдельные характеристики продукции. Полученные с его помощью результаты позволяют определить удовлетворенность пользователя от использования тех или иных функциональных возможностей (в нашем случае методов построения и распознавания маршрута).

Анкета Кано [75, 76] задает по 2 связанных вопроса — в позитивном и негативном ключе, например:

1. Насколько вам понравится наличие такой характеристики в продукте?
2. Как вы отнесетесь к тому, что эта характеристика будет выражена слабо или же будет отсутствовать вообще?

Варианты ответов для каждого из двух вопросов (аналогичные ответы использовались для описанных далее анкет):

1. Мне это нравится;
2. Я ожидаю, что эта характеристика будет в продукте;
3. Я отношусь к ней нейтрально;
4. Я могу ее терпеть;
5. Мне это не нравится.

Для определения категории реакции пользователей с помощью их ответов на 2 вопроса Кано разработал специальную оценочную таблицу (рис. 37).

Положительный вопрос	Отрицательный вопрос				
	НРАВИТСЯ	ОЖИДАЮ	НЕЙТРАЛЬНО	МОГУ ТЕРПЕТЬ	НЕ НРАВИТСЯ
НРАВИТСЯ	-	Привлекательные	Привлекательные	Привлекательные	Одномерные
ОЖИДАЮ	Нежелательные	Неважные	Неважные	Неважные	Обязательные
НЕЙТРАЛЬНО	Нежелательные	Неважные	Неважные	Неважные	Обязательные
МОГУ ТЕРПЕТЬ	Нежелательные	Неважные	Неважные	Неважные	Обязательные
НЕ НРАВИТСЯ	Нежелательные	Нежелательные	Нежелательные	Нежелательные	-

Рис. 37: Оценочная таблица Кано

В примере, приведенном на рис. 37, выделенная строка показывает ответ на первый (положительный) вопрос. Выделенный столбец показывает ответ на второй (отрицательный) вопрос. Пересечение строки и столбца показывает тип категории для этой характеристики (в данном случае тип «Привлекательные»). Одномерные характеристики (выделены оранжевым цветом на рисунке) вызывают удовлетворение (если они есть) или неудовлетворение (если их нет). Такая линейная зависимость между сложностью характеристики и эмоциональной реакцией верна, прежде всего, для базовых характеристик, таких как: простота использования, стоимость, важность.

Далее рассмотрим предлагаемые опрашиваемым анкеты. С целью экономии места варианты возможных ответов на вопросы были опущены там, где они подразумеваются из здравого смысла.

### 3.9.1 Построение описания маршрута

Для сравнения методов построения описания маршрутов были выработаны следующие вопросы:

1. общие вопросы касательно пола, возраста и т.д.;

2. вступительные вопросы:

- a. Как часто вы пользуетесь навигационно-картографическими системами?
- b. Какими навигационно-картографическими системами Вы пользуетесь чаще всего?
- c. Пользуетесь ли Вы функцией прокладывания маршрута?

3. основные вопросы:

- a. Как часто Вы пользуетесь функцией описания маршрута при помощи изображения траектории для изучения и запоминания маршрута? (к вопросу были приведены иллюстрации)
- b. Как часто Вы пользуетесь функцией описания маршрута при помощи пошаговой инструкции к перемещению для изучения и запоминания маршрута? (к вопросу были приведены иллюстрации)

4. вопросы по методу Кано:

- a. в положительном ключе:
  - i. Навигационно-картографические системы предлагают текстовое описание маршрута. (к вопросу были приведены иллюстрации)
  - ii. Текстовое описание маршрута будет дополнять существующие способы получения представления о маршруте.
  - iii. Текстовое описание маршрута учитывает мои персональные знания и подстраивается лично под меня.
  - iv. Персональный робот умеет голосом озвучивать описание маршрута.

b. в негативном ключе:

- i. Навигационно-картографические системы не предлагают текстовое описание маршрута. (*к вопросу были приведены иллюстрации*)
- ii. Текстовое описание маршрута будет единственным способом получить представление о маршруте.
- iii. Текстовое описание маршрута не учитывает мои персональные знания и не подстраивается лично под меня.
- iv. Персональный робот не умеет голосом озвучивать описания маршрута.

Статистика по общим вопросам анкеты:

- было опрошено 270 человек;
- 98 женщин и 172 мужчины;
- возраст испытуемых — от 25 до 32 лет.

Обработанные результаты опроса выглядят следующим образом:

- вступительные вопросы:
  - частота использование навигационно-картографических систем: *несколько раз в неделю*;
  - используемые навигационные системы: *Яндекс Карты, Google Maps*;
  - использование функции прокладывания маршрута: *да*;
- основные вопросы:
  - частота использования изображения траектории: *каждый раз при прокладывании маршрута*;

- частота использования последовательности действий к перемещению: *очень редко*;
- вопросы по методу Кано:
  - вопрос i: *привлекательно*
  - вопрос ii: *одномерная характеристика*
  - вопрос iii: *обязательно*
  - вопрос iv: *неважно*

Результаты показывают, что появление текстового описания в навигационно-картографических системах очень ожидаемо и полноправно соперничает с существующими способами описания пути.

### **3.9.2 Распознавание описания маршрута**

Для сравнения методов распознавания описания маршрутов были выработаны следующие вопросы:

1. общие вопросы касательно пола, возраста и т.д.;
2. вступительные вопросы:
  - a. Как часто вы пользуетесь навигационно-картографическими системами?
  - b. Какими навигационно-картографическими системами Вы пользуетесь чаще всего?
  - c. Пользуетесь ли Вы функцией ввода собственного маршрута (чтобы узнать время движения по нему с учетом пробок, чтобы узнать возможность подобного проезда с учетом дорожной разметки и т.д.)?
3. основные вопросы:
  - a. Как часто Вы пользуетесь функцией ввода маршрута при помощи задания опорных точек?

#### 4. вопросы по методу Кано:

##### а. в положительном ключе:

- i. На мобильных платформах есть возможность ввода конкретного маршрута.
- ii. На мобильных платформах есть возможность ввода маршрута голосом или текстом при помощи персонального ассистента («Siri» или «ОК Google»).
- iii. В навигационно-картографических системах появляется возможность вводить конкретный маршрут, описывая его текстом, при этом остается способ его задания через опорные точки.
- iv. Текстовый ввод маршрута учитывает мои персональные знания и подстраивается лично под меня.
- v. Персональный робот умеет воспринимать голосовое описание маршрута и, например, следовать по нему.

##### б. в негативном ключе:

- i. На мобильных платформах нет возможности ввода конкретного маршрута.
- ii. На мобильных платформах нет возможности ввода маршрута голосом или текстом при помощи персонального ассистента («Siri» или «ОК Google»).
- iii. В навигационно-картографических системах нет возможности вводить конкретный маршрут,

описывая его текстом, существует только способ его задания через опорные точки.

iv. Текстовый ввод маршрута не учитывает мои персональные знания и не подстраивается лично под меня.

v. Персональный робот не умеет воспринимать голосовое описание маршрута и, соответственно, следовать по нему.

Статистика по общим вопросам анкеты:

- было опрошено 270 человек;
- 98 женщин и 172 мужчины;
- возраст испытуемых — от 25 до 32 лет.

Обработанные результаты опроса выглядят следующим образом:

- вступительные вопросы:
  - частота использования навигационно-картографических систем: *несколько раз в неделю*;
  - используемые навигационные системы: *Яндекс Карты, Google Maps*;
  - использование функции ввода маршрута: *да*;
- основные вопросы:
  - частота использования ввода маршрута при помощи опорных точек: *редко*;
- вопросы по методу Кано:
  - вопрос i: *привлекательно*
  - вопрос ii: *неважно*
  - вопрос iii: *обязательно*

- вопрос iv: *обязательно*
- вопрос v: *неважно*

Результаты показывают, что пользователи заинтересованы в новом способе ввода маршрута, но пока не готовы отказаться от существующих вариантов.

### 3.9.3 Использование системы навигации в офисе SkyLight

Для изучения популярности использования системы описания пути к рабочему месту сотрудника, предлагаемой на корпоративном портале компании Mail.Ru внутри офиса SkyLight, была собрана статистика по использованию и проведен мини опрос.

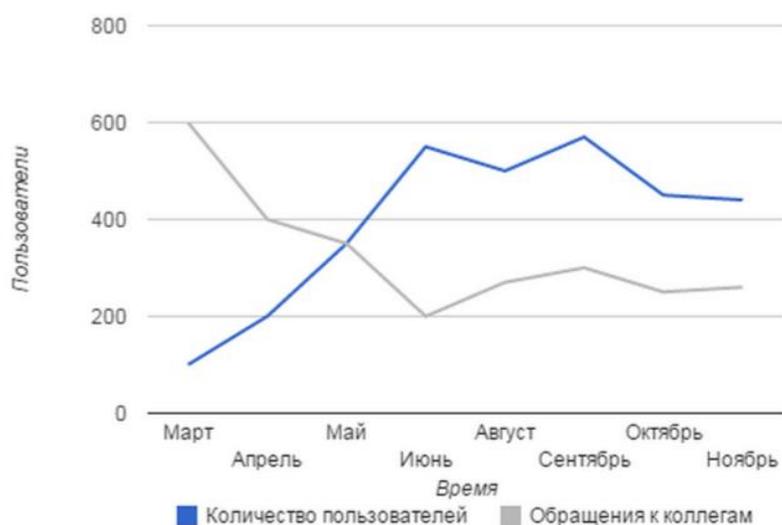


Рис. 38: Количество пользователей системы описания маршрута к рабочему месту сотрудника в офисном здании SkyLight

Статистика по использованию основывается на количестве нажатий на кнопку «как пройти?», расположенную непосредственно рядом с указанием адреса рабочего места сотрудника.

Опрос проводился один раз в месяц и состоял из одного вопроса «Спрашивали ли у Вас за последний месяц о том, как пройти к какому-либо сотруднику?».

Результаты собранной статистики и проведенных опросов отображены ниже на графике (рис. 38). Запуск системы был проведен в марте, исследование велось 8 месяцев. По результатам видно, что в течение первых 4 месяцев пользователи привыкали к новому функционалу и продолжали по-прежнему спрашивать коллег на этаже дорогу к нужному сотруднику. Через полгода большинство сотрудников перешли на использование системы и больше не беспокоили коллег уточняющими вопросами.

### **3.10 Выводы к третьей главе**

В третьей главе детально рассмотрена архитектура и реализация программного обеспечения, имплементирующего разработанные алгоритмы построения и распознавания текстового описания. Были рассмотрены различные фреймворки и языки программирования, выявлены их преимущества и недостатки. В главе подробно описана методика сравнения предложенных алгоритмов с существующими решениями с использованием метода Кано и приведена успешная статистика использования разработанных алгоритмов на практике. На основе собранных данных можно сделать вывод о востребованности описанной разработки в частности и самой задачи по построению и распознаванию описаний маршрутов в общем.

## Заключение

Основные результаты работы:

1. На основе проведенного исследования методов ввода и вывода маршрута разработаны и реализованы новые алгоритмы построения и распознавания персонализированного когнитивного текстового описания маршрута, упрощающие использование навигационных систем.
2. На основе проведенных исследований и экспериментов по формированию когнитивной карты в сознании человека и восприятию текстового описания пути разработаны алгоритмы для хранения и отображения данных о маршруте, учитывающие психологию восприятия картографической информации.
3. Предложена методика проверки качества предлагаемых текстовых описаний и их удобства для человека на основе статистики использования и анкетирования, выявляющая объективное мнение пользователей.
4. На основе разработанных методов и алгоритмов реализована программная система, интегрируемая в существующие картографические системы в качестве подключаемого модуля и используемая в реальных практических приложениях.

## Литература

1. G. Menshikova, Yu. Bayakovski, E. Luniakova, M. Pestun, D. Zakharkin, Virtual Reality Technology for the Visual Perception Study, Springer //Transactions on Computational Science XIX, Lecture Notes in Computer Science Volume 7870. - Germany, 2013. P. 107-116.
2. G. Menshikova, Yu. Bayakovski, E. Luniakova, M. Pestun, D. Zakharkin, Virtual reality technology for the visual perception study //Proc. of the 22nd International Conference on Computer Graphics and Vision (GraphiCon'2012). - Moscow, Russia, 2012. P. 51-54.
3. М.В. Пестун, Компьютерная система описания маршрута в удобном для человека формате //Научно-практический семинар "Новые информационные технологии в автоматизированных системах". - М., Россия, 2014. С. 125-134.
4. М.В. Пестун, Методы преобразования текстового описания маршрута в компьютерное представление //Научно-практический семинар "Новые информационные технологии в автоматизированных системах". - М., 2015. С. 510-517.
5. A. Tetereva, M. Pestun, G. Menshikova, Effect of negative emotions on the cognitive maps acquisition //Proc. of 37 th European Conference on Visual Perception. - Perception, v.43, ECVP Abstract supplement. - Belgrad, Serbia, 2014. P. 140.
6. Г.Я. Меньшикова, Ю.М. Баяковский, Е.Г. Лунякова, М.В. Пестун, Д.В. Захаркин, Эффект артикуляции в трехмерных зрительных иллюзиях //Экспериментальная психология. - М.: ГБОУ ВПО «Московский городской психолого-педагогический университет», 2013, №2. С. 46-57.
7. М.В. Пестун, Алгоритмы построения и хранения навигационной когнитивной карты для взаимодействия с человеком //Труды 24-й

международной конференции по компьютерной графике и зрению  
ГрафиКон'2014. - Ростов н/Д, 2014. С. 119-122.

8. М.В. Пестун. Когнитивная навигация и алгоритм построения  
текстового описания маршрута в удобном для человека виде  
//Программные продукты и системы /Гл. ред. академик РАН С.В.  
Емельянов. - Тверь: Научно-исследовательский институт  
«Центрпрограммсистем», 2015. С. 28-33.
9. В.А. Галактионов, М.В. Пестун, Алгоритмы построения и  
распознавания навигационных описаний маршрутов для  
картографических компьютерных систем //Труды 25-й  
международной конференции по компьютерной графике и зрению  
ГрафиКон'2015. - Протвино, 2015. С. 62-67.
10. Allen G.L., Kirasic K.C., Siegel A.W., and Herman J.F., Development  
Issues in Cognitive Mapping: The Selection and Utilization of  
Environmental Landmarks //Child Development, 1979, no. 50. P.  
1062-1070.
11. Loomis J.M., Klatzky R.L., and Colledge R.G., Human navigation by  
path integration //Wayfinding: Cognitive mapping and spatial behavior,  
1999.
12. Klatzky R.L., Freksa C., Habel C., and Wender K.F., Allocentric and  
Egocentric Spatial Representations: Definitions, Distinctions, and  
Interconnections //Artificial Intelligence (Eds.), Springer, 1997, vol.  
1404. P. 1-17.
13. Klatzky R.L., Loomis J.M., Golledge R.G., Cicinelli J.G., Doherty S.,  
and Pellegrino J.W., Acquisition of route and survey knowledge in the  
absence of vision //Journal of Motor Behavior, 1990, No. 22. P. 19-43.
14. Lakhtionova I., Menshikova G. The method of testing the ability of  
allocentric cognitive maps acquisition // Proc. of 36 th European  
Conference on Visual Perception. Bremen, Germany, 2013. P. 53.

15. Cousins J.H., Siegel A.W., and Maxwell S.E., Way Finding And Cognitive Mapping in Large Scale Environments: A Test of a Developmental Model //Journal of Experimental Child Psychology. 1983. No. 35. P. 1-20.
16. W.K. Yeap, M.E. Jefferies, On early cognitive mapping. Spatial Cognition and Computation //The Netherlands: Kluwer Academic Publishers, vol. 2 (2), 2000. P. 85-116.
17. Xiujun Ma, Zhongya Wei, Yanwei Chai, Kunqing Xie, Integrating Map Services and Location-based Services for Geo-Referenced Individual Data Collection //Geoscience and Remote Sensing Symposium. - IGARSS. - IEEE International, vol.5. P. 7-11.
18. Global Positioning System  
URL: [http://en.wikipedia.org/wiki/Global\\_Positioning\\_System](http://en.wikipedia.org/wiki/Global_Positioning_System) (дата обращения 10.05.2015)
19. ГЛОНАСС  
URL: <http://en.wikipedia.org/wiki/GLONASS> (дата обращения 10.05.2015)
20. BeiDou Navigation Satellite System  
URL: [http://en.wikipedia.org/wiki/BeiDou\\_Navigation\\_Satellite\\_System](http://en.wikipedia.org/wiki/BeiDou_Navigation_Satellite_System) (дата обращения 10.05.2015)
21. M. Visvalingam, Trends and concerns in digital cartography //Computer-Aided Design, vol. 22, issue 3, 1990, P. 115-130.
22. E.C. Tolman, Cognitive maps in rats and men //Psychological review, 55(4), 1948. P. 189.
23. Navitel Navigation System  
URL: <http://navitel.ru/> (дата обращения 10.05.2015)
24. Garmin Navigation System  
URL: <http://www.garmin.com/> (дата обращения 10.05.2015)
25. Mio Navigation System  
URL: <http://www.mio.com/> (дата обращения 10.05.2015)

26. Apple Maps for iOS  
URL: <https://www.apple.com/ios/maps/> (дата обращения 10.05.2015)
27. Google Maps for Android  
URL:  
<https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en> (дата обращения 10.05.2015)
28. Яндекс Карты для мобильных устройств  
URL: <https://mobile.yandex.ru/apps/maps/android/> (дата обращения 10.05.2015)
29. Яндекс Навигатор для мобильных устройств  
URL: <https://mobile.yandex.ru/apps/navi/android/> (дата обращения 10.05.2015)
30. Android Wear  
URL: <http://www.android.com/wear/> (дата обращения 10.05.2015)
31. Apple Watch  
URL: <https://www.apple.com/watch/> (дата обращения 10.05.2015)
32. Google Glass  
URL: <http://www.google.ru/glass/start/> (дата обращения 10.05.2015)
33. Google Maps for Web Browser  
URL: <https://www.google.ru/maps> (дата обращения 10.05.2015)
34. Яндекс Карты для веб-браузера  
URL: <https://maps.yandex.ru/> (дата обращения 10.05.2015)
35. Bing Maps for Web Browser  
URL: <http://www.bing.com/maps/> (дата обращения 10.05.2015)
36. Calculate distance, bearing and more between Latitude/Longitude points  
URL: <http://www.movable-type.co.uk/scripts/latlong.html> (дата обращения 10.05.2015)
37. Д. Э. Розенталь, Справочник по правописанию и стилистике. - М.: Комплект, 1997.

38. Е. Калачинская, Практическая грамматика русского языка. - Владивосток: ВГУЭС, 2015.
39. А. А. Зализняк, Грамматический словарь русского языка. - М.: Словоизменение, 1980.
40. Л. А. Введенская, Л. Г. Павлова, Е. Ю. Кашаева, Русский язык. Практикум: учебное пособие для вузов. /ЛА Введенская, МЮ Семёнова. - М., 2009.
41. А. А. Камынина, Современный русский язык: морфология. – М.: Издательство Московского университета, 2010.
42. Как склонять географические названия?  
URL: [http://www.gramota.ru/spravka/letters/?rub=rubric\\_90](http://www.gramota.ru/spravka/letters/?rub=rubric_90) (дата обращения 10.05.2015)
43. Siri  
URL: <https://www.apple.com/ios/siri/> (дата обращения 10.05.2015)
44. OK Google  
URL: <https://www.google.com/okgoogle> (дата обращения 10.05.2015)
45. Jiang Guo, An approach for modeling and designing software architecture //Engineering of Computer-Based Systems. - Proceedings. 10th IEEE International Conference and Workshop, 2003. P. 89-97.
46. H.M. Sneed, C. Verhoef, S.H. Sneed, Reusing existing object-oriented code as web services in a SOA, Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA) //IEEE 7th International Symposium, 2013. P. 31-39.
47. M. Tanaka, T. Kume, A. Matsuo, Web API Creation for Enterprise Mashup // Services, IEEE World Congress, 2011. P. 319-326.
48. Yoon-Jung Rhee, Chang-Won Choi, Tae-Woo Kim, Tai-Yun Kim, Client-side mechanism for improving busy Web server's performance //Info-tech and Info-net. - Proceedings. ICII. - Beijing, International Conferences, vol.5, 2001. P. 95-100.

49. A. Ribeiro, A.R. da Silva, Survey on Cross-Platforms and Languages for Mobile Apps //Quality of Information and Communications Technology (QUATIC). - Eighth International Conference, 2012. P. 255-260.
50. Java Virtual Machine  
URL: [http://en.wikipedia.org/wiki/Java\\_virtual\\_machine](http://en.wikipedia.org/wiki/Java_virtual_machine) (дата обращения 10.05.2015)
51. C# Programming Language  
URL: [http://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (дата обращения 10.05.2015)
52. JavaScript Programming Language  
URL: <http://en.wikipedia.org/wiki/JavaScript> (дата обращения 10.05.2015)
53. PHP Programming Language  
URL: <http://en.wikipedia.org/wiki/PHP> (дата обращения 10.05.2015)
54. Python Programming Language  
URL: [http://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Python_(programming_language)) (дата обращения 10.05.2015)
55. Карты Mail.Ru  
URL: <http://maps.mail.ru>
56. Dropbox Online File Storage  
URL: <http://www.dropbox.com>
57. Box Online File Storage  
URL: <http://www.box.com> (дата обращения 10.05.2015)
58. Cloud Mail.Ru Online File Storage  
URL: <http://cloud.mail.ru> (дата обращения 10.05.2015)
59. Foursquare Social Network  
URL: <http://www.foursquare.com> (дата обращения 10.05.2015)
60. Facebook Social Network  
URL: <http://www.facebook.com> (дата обращения 10.05.2015)

61. Vkontakte Social Network  
URL: <http://www.vk.com> (дата обращения 10.05.2015)
62. Haibo Zhao, P. Doshi, Towards Automated RESTful Web Service Composition //Web Services, 2009. - ICWS 2009. IEEE International Conference, 2009. P.189-196.
63. HTML 5 Specification  
URL: <http://www.w3.org/html/wg/drafts/html/master/> (дата обращения 10.05.2015)
64. Web Protocols Specifications  
URL: <http://www.w3.org/Protocols/> (дата обращения 10.05.2015)
65. jQuery  
URL: <https://jquery.com/> (дата обращения 10.05.2015)
66. KendoUI  
URL: <http://www.telerik.com/kendo-ui> (дата обращения 10.05.2015)
67. AngularJS  
URL: <https://angularjs.org/> (дата обращения 10.05.2015)
68. Bootstrap  
URL: <http://getbootstrap.com/> (дата обращения 10.05.2015)
69. Foundation  
URL: <http://foundation.zurb.com/> (дата обращения 10.05.2015)
70. Semantic  
URL: <http://semantic-ui.com/> (дата обращения 10.05.2015)
71. Pure  
URL: <http://purecss.io/> (дата обращения 10.05.2015)
72. UIKit  
URL: <http://getuikit.com/> (дата обращения 10.05.2015)
73. GitHub Online Source Code Storage System  
URL: <http://www.github.com> (дата обращения 10.05.2015)
74. Google Drive Online File Storage  
URL: <http://drive.google.com> (дата обращения 10.05.2015)

75. Описание метода «Модель Кано»  
URL: <http://www.inventech.ru/pub/methods/metod-0022/> (дата обращения 10.05.2015)
76. Модель Кано  
URL: <http://www.fdfgroup.ru/?id=281> (дата обращения 10.05.2015)
77. Yu-Cheng Lee; Ya-Li Wang; Shao-Bin Lin, The reformed analytical Kano model //Service Systems and Service Management (ICSSSM). - 9th International Conference, 2012. P. 676-679.
78. Sauerwein, Elmar, Franz Bailom, Kurt Matzler, Hans H. Hinterhuber, The Kano model: How to delight your customers //In International Working Seminar on Production Economics, vol. 1, 1996. P. 313-327.
79. О.И. Давыдов, А.К. Платонов, Сеть Пассфреймов —  
комбинированная модель операционной среды мобильного робота  
//Препринт ИПМ № 15. - М.: ИПМ им. М.В. Келдыша РАН, 2015.
80. Paillard J., The motor determinants of spatial organization //Cahiers de Psychologie, 1971, vol. 14. P. 261 316.

## Список рисунков

Рис. 1: Пример экрана системы Navitel.....	19
Рис. 2: Пример экрана системы Garmin .....	19
Рис. 3: Пример экрана системы Mio.....	20
Рис. 4: Пример экрана системы Apple Maps (изображение траектории) .....	22
Рис. 5: Пример экрана системы Apple Maps (пошаговая инструкция к перемещению).....	23
Рис. 6: Пример экрана системы Google Maps (изображение траектории и пошаговая инструкция к перемещению) .....	24
Рис. 7: Пример экрана системы Яндекс Карты (изображение траектории).....	24
Рис. 8: Пример экрана системы Яндекс Навигатор (изображение траектории).....	25
Рис. 9: Пример пользовательского интерфейса системы Google Wear .....	26
Рис. 10: Пример пользовательского интерфейса системы Apple Watch .....	27
Рис. 11: Пример пользовательского интерфейса (того, что видит пользователь с надетым на голову устройством) системы Google Glass .....	28
Рис. 12: Стационарный терминал в торговом центре с сенсорным экраном.....	29
Рис. 13: Стационарный терминал в торговом центре без сенсорного экрана .....	30
Рис. 14: Пример пользовательского интерфейса системы Google Maps .....	31
Рис. 15: Пример пользовательского интерфейса системы Яндекс Карты .....	31
Рис. 16: Пример пользовательского интерфейса системы Bing Maps .....	32
Рис. 17: Описание маршрута в виде пошаговой инструкции к перемещению в навигационно-картографических системах Bing Maps, Google Maps, Яндекс Карт (слева направо) .....	34
Рис. 18: Описание маршрута в виде изображения траектории пути в навигационно-картографических системах Google Maps, Garmin, iGo, Яндекс Навигатор (слева направо, сверху вниз) .....	35
Рис. 19: Пример введенного маршрута при помощи задания опорных точек в онлайн сервисе Яндекс Карты .....	37

Рис. 20: Схематическое представление системы построения виртуальной реальности CAVE.....	39
Рис. 21: Визуальное представление первых трех шагов на схеме маршрута [16]..	43
Рис. 22: Поиск нужной точки на пути для получения фотографии POI от навигационно-картографического сервиса.....	52
Рис. 23: Схема работы всего алгоритма.....	54
Рис. 24: Схема архитектуры программного решения.....	79
Рис. 25: Схема изменения адаптивного пользовательского интерфейса под размеры экрана .....	82
Рис. 26: Автоматическое дополнение названий POI .....	85
Рис. 27: Интерфейс пользователя для задания маршрута и получения текстового описания .....	86
Рис. 28: Интерфейс пользователя для ввода текстового описания маршрута .....	87
Рис. 29: Пример перевода изображения траектории пути в текстовое представление с использованием знакомых пользователю маршрутов и POI.....	88
Рис. 30: Составные части текстового описания пути, используемые алгоритмом	88
Рис. 31: Автоматическое дополнение при описании маршрута.....	89
Рис. 32: Многофункциональный робот РБ-2.....	90
Рис. 33: Пример построенной роботом карты окружения с использованием технологии пассфреймов.....	91
Рис. 34: Автономный квадрокоптер, участвовавший в конкурсе летающих роботов от компании КРОК.....	92
Рис. 35: Пример перевода последовательности действий к перемещению, генерируемых навигационно-картографической системой Карты Mail.Ru в ее текстовое описание .....	93
Рис. 36: Пример текстового описания пути к рабочему месту сотрудника с началом от лифтового холла внутри офисного здания Skylight .....	95
Рис. 37: Оценочная таблица Кано.....	97
Рис. 38: Количество пользователей системы описания маршрута к рабочему месту сотрудника в офисном здании SkyLight .....	103