

# Содержание

<b>1</b>	<b>Конструкторская часть</b>	<b>2</b>
1.1	Разработка технического задания . . . . .	2
1.1.1	Постановка задачи проектирования . . . . .	2
1.1.2	Описание предметной области . . . . .	2
1.1.3	Выбор критериев качества . . . . .	12
1.1.4	Анализ аналогов и прототипов . . . . .	13
1.1.5	Перечень задач, подлежащих решению в процессе раз- работки . . . . .	21
1.2	Проектирование подсистемы . . . . .	22
1.2.1	Разработка структуры подсистемы . . . . .	22
1.2.2	Разработка формата и структуры данных . . . . .	36
1.2.3	Разработка алгоритмов обработки информации . . . . .	38

# **1 Конструкторская часть**

## **1.1 Разработка технического задания**

### **1.1.1 Постановка задачи проектирования**

Целью разработки подсистемы автономного определения перемещения объекта является предоставления удобного с точки зрения интеграции компонента для встраивания во многие бытовые автономные автоматические системы, в то же время дешевого и не требующего специализированных устройств для своей работы.

### **1.1.2 Описание предметной области**

#### **1.1.2.1 Естественнo-языковое описание процесса.**

В ходе создания подвижных автономных систем возникает задача определения текущего местоположения объекта в пространстве.

Для этого можно использовать различные методы, основанные на глобальном позиционировании в географической системе координат с использованием позиционирования по спутникам (GPS, ГЛОНАСС), или методы, основанные на определении перемещения от стартовой позиции. При этом у данных методов разные сферы применения. Так, например, при позиционировании внутри помещения использование спутниковых систем позиционирования становится невозможным по причине слабого сигнала или его полного отсутствия, а так же из-за недостаточной точности в рамках навигации внутри интерьера помещения. Так же не стоит забывать про актуальность систем навигации в космической отрасли, где использование спутников является невозможным в принципе. Вторую группу методов принято называть методами одометрии, которые могут быть основаны на:

- на вращении колес;
- использовании инерциальных измерительных приборов;
- компьютерном зрении.

Каждый из них обладает своими плюсами и минусами [1], но развитие вычислительной техники и алгоритмов компьютерного зрения дало мощный толчок к более широкому применению визуальной одометрии. Данный подход позволяет получать один видеоряд через видеокамеру и на его основе получать разные сведения об окружающей среде. Тем не менее он не лишен недостатков. Для борьбы с ними применяется комбинация нескольких методов одометрии.

Одним из вариантов соещения методов является использование метода визуальной одометрии и инерционных измерительных устройств.

При таком гибридном методе данные с видеокамеры и данные с инерционных измерительных устройств обрабатываются параллельно и независимо. В результате получаются два независимых рассчитанных положения носителя, после чего они сопоставляются, и из них выбираются наиболее правдоподобные.

Таким образом, в процессе функционирования спроектированного модуля визуальной одометрии происходит следующий бесконечный процесс. На вход модуля непрерывно подается видео поток и данные об угловых скоростях и ускорении объекта относительно трех взаимноперпендикулярных осей. Эти данные обрабатываются параллельно в соответствующих модулях, на выходе каждого из которых получаем смещение объекта относительно предыдущего положения и его поворот. Далее эти данные совмещаются и выбираются наиболее правдоподобные, которые затем прибавляются к положению и углу поворота, высчитанным на предыдущей итерации.

#### **1.1.2.2 Графическое представление процесса**

Графическое представление процесса представлено на рисунке 1.

#### **1.1.2.3 Вычисление оптического потока.**

**Оптический поток** — это изображение видимого движения объектов, поверхностей или краев сцены, получаемое в результате перемещения наблюдателя (глаз или камеры) относительно сцены[2].

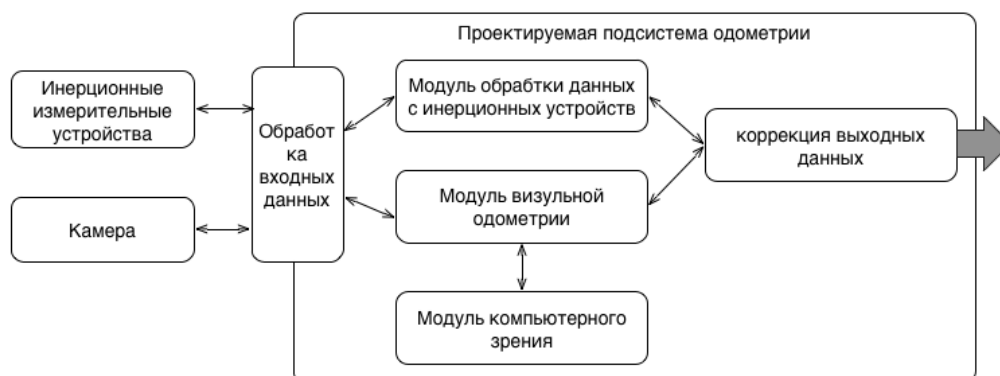


Рис. 1: Графическое представление процесса.

Существует несколько подходов к определению смещений между двумя соседними кадрами. Например, можно для каждого небольшого фрагмента (скажем, 8 на 8 пикселей) одного кадра найти наиболее похожий фрагмент на следующем кадре. В этом случае разность координат исходного и найденного фрагментов даст нам смещение. Основная сложность тут состоит в том, как быстро отыскать нужный фрагмент, не перебирая весь кадр пиксель за пикселем. Различные реализации этого подхода так или иначе решают проблему вычислительной сложности. Некоторые настолько успешно, что применяются, например, в распространенных стандартах сжатия видео. Платой за скорость естественно является качество. Мы же рассмотрим другой подход, который позволяет получить смещения не для фрагментов, а для каждого отдельного пикселя, и применяется тогда, когда скорость не столь критична. Именно с ним в литературе часто связывают термин “оптический поток”.

Данный подход часто называют дифференциальным, поскольку в его основе лежит вычисление частных производных по горизонтальному и вертикальному направлениям изображения. Как мы увидим далее, одних только производных недостаточно чтобы определить смещения. Именно поэтому на базе одной простой идеи появилось великое множество методов, каж-

дый из которых использует какую-нибудь свою математическую пляску с бубном, чтобы достичь цели. Сконцентрируемся на методе Лукаса-Канаде (Lucas-Kanade), предложенном в 81 году Брюсом Лукасом и Такео Канаде. Данный метод является наименее ресурсоемким [3] при этом обеспечивает приемлемое качество вычисления оптического потока.

С математической точки зрения данный алгоритм можно описать следующим образом. Пусть даны два изображения  $F1$  и  $F2$ , и нам требуется найти смещение точки с координатой  $x$ . Рассматривая два последовательных изображения можно сказать:

$$f_2(x) = f_1(x - d)$$

Обратите внимание, что  $f_1$  и  $f_2$  при желании можно записать и в общем виде:  $f_1(x) = I(x, y, t)$  ;  $f_2(x) = I(x, y, t + 1)$ .

Свяжем известные значения со смещением  $d$ . Для этого запишем разложение в ряд Тейлора для  $f_1(x - d)$ :

$$f_1(x - d) = f_1(x) + df'_1(x) + O(d^2 f''_1)$$

Предположим, что  $f_1(x - d)$  достаточно хорошо аппроксимируется первой производной. Сделав это предположение, отбросим всё что после первой производной:

$$f_1(x - d) = f_1(x) + df'_1(x)$$

Смещение  $d$  — это наша искомая величина, поэтому необходимо преобразовать  $f_1(x - d)$  . Как мы условились ранее,  $f_2(x) = f_1(x - d)$ , поэтому просто перепишем:

$$f_2(x) = f_1(x) - df'_1(x)$$

Отсюда следует:

$$d = \frac{f_1(x) - f_2(x)}{f'_1(x)}$$

Следует отметить, что выше был рассмотрен одномерный случай и были сделаны несколько грубых допущений. Но описание алгоритма Лукаса-Канаде для двумерного случая только усложняет математические выводы и понимание сути.

Для снижения погрешности вызванной отбрасыванием старших производных смещение для каждой пары кадров (назовём их  $F_i$  и  $F_{i+1}$ ) можно вычислять итеративно. В литературе это называется искажением (warping). На практике это означает, что, вычислив смещения на первой итерации, мы перемещаем каждый пиксель кадра  $F_{i+1}$  в противоположную сторону так, чтобы это смещение компенсировать. На следующей итерации вместо исходного кадра  $F_{i+1}$  мы будем использовать его искаженный вариант  $F_{i+1}^1$ . И так далее, пока на очередной итерации все полученные смещения не окажутся меньше заданного порогового значения. Итоговое смещение для каждого конкретного пикселя мы получаем как сумму его смещений на всех итерациях [4].

Так же следует отметить, что данный алгоритм плохо работает на однотонных изображениях. Данный недостаток является самым критичным.

#### **1.1.2.4 Одометрия с использованием инерциальных измерительных устройств**

Навигационные решения надлежащего качества могут быть получены именно в результате взаимодействия или последующей совместной обработки данных от двух источников - визуальной одометрии и инерциальной системы.

В наиболее общей форме можно определить инерциальную систему как ортогональную триаду гироскопов и акселерометров, выполняющих непосредственные геопространственные измерения и вычислительный блок, осуществляющий алгоритмические преобразования данных непосредственных измерений.

Следует отметить, что гироскоп любого типа позволяет определять ориентацию в геодезическом пространстве в любой момент времени независимо от местоположения, скорости и других параметров носителя. Точность поставляемых гироскопом данных во всех случаях подвержена деградации («ухода») с течением времени. Величина «ухода» значительна и может составлять до нескольких градусов в час [5].

Акселерометры предназначены для измерения линейных ускорений. В равной степени они пригодны для измерений сил, так как согласно ньютоновской механике сила и ускорение есть разные проявления одного и того

же физического явления.

В общем случае в системах навигации следует определять следующие показатели:

- **Рыскание** — угловые движения летательного аппарата, судна, автомобиля относительно вертикальной оси (см. также вертикальная ось самолёта), а также небольшие изменения курса вправо или влево, свойственные судну [6];
- **Крен** — поворот объекта (судна, самолёта, фундамента) вокруг его продольной оси [7];
- **Тангаж** — угловое движение летательного аппарата или судна относительно главной (горизонтальной) поперечной оси инерции [8].

С учетом сделанных замечаний рассмотрим основные процедуры, выполняемые в навигационном комплексе на базисном уровне.

#### ***Вычисление крена и тангажа посредством акселерометров***

Обладая чувствительностью к земной гравитации, акселерометры обеспечивают измерение долговременных значений крена и тангажа по схеме, изображенной на рисунке 2. Рассмотрим акселерометр, рабочая ось которого совпадает со строительной осью  $OX$  носителя.

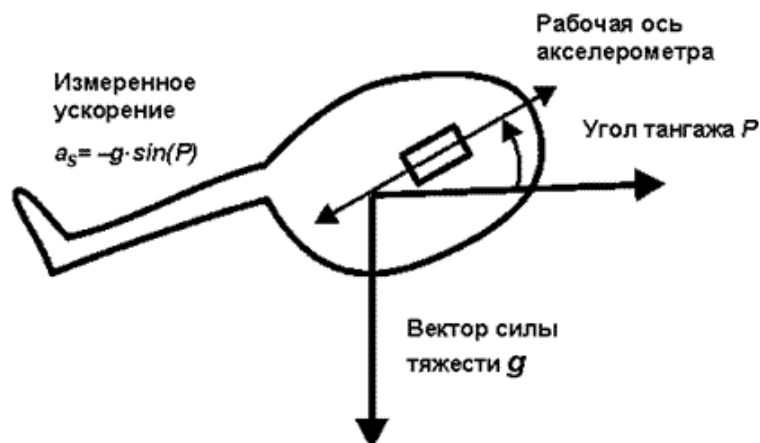


Рис. 2: Измерения величин крена и тангажа посредством акселерометров.

Полагая ускорение носителя равным нулю, мы можем вычислить угол тангажа как:

$$P = \arcsin(-a_s/g)$$

Аналогично вычисляется угол крена. Таким образом, два из трех углов, определяющих угловую ориентацию, могут быть определены только за счет использования акселерометров. Это совершенно очевидный результат, принимая во внимание то обстоятельство, что углы крена и тангажа по изначальным определены по отношению к вертикали, которая в нашем случае соответствует вектору тяжести. Однако, здесь следует признать, что описанный метод не может быть использован на практике сам по себе, так как в описанной схеме существенно состояние покоя, в котором должна находиться система. Если это условие не соблюдается, то совершенно очевидно, что отсутствует принципиальная возможность выделить вектор ускорения свободного падения из суммы всех ускорений, которую испытывает система.

### ***Вычисление изменений ориентации с использованием гироскопов***

Как отмечено выше, в конструкции навигационного комплекса используются оптические гироскопы, обладающие чувствительностью к изменениям ориентации т.е. к величине угловой скорости. Интегрирование (численное суммирование) значений, измеренных гироскопами, обеспечивает определение кратковременных угловых перемещений в физическом пространстве.

Для корректного расчета угла поворота должны быть учтены внутренние ошибки гироскопа – дрейф, ошибка масштабного коэффициента, случайный шум.

При этом внутренние ошибки гироскопа полностью смешаны с истинными значениями и не могут быть отделены от них на базисном информационном уровне. В процессе дальнейшей обработки эта смесь подвергается интегрированию, в результате чего возникает ошибочное угловое смещение, которое, таким образом, приобретает долговременный характер (рис. 3). Точная оценка величины ошибочного углового смещения и его устранение осуществляется при генерации навигационного решения на последующем навигационном уровне.

### ***Определение координат пространственного положения с помощью акселерометров.***

Наличие акселерометров позволяет определять величины линейных ускорений, которые испытывает система. Положим, что ориентация систе-



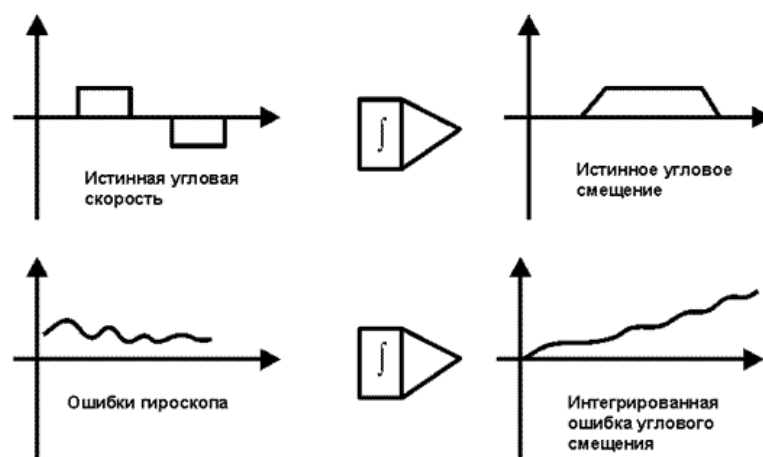


Рис. 3: Схема определения углового смещения.

мы в физическом пространстве определена точно с помощью методов, описанных выше. Тогда имеется возможность выделить вектор силы гравитации среди всей суммы векторов сил, приложенных к системе и, следовательно, оценить величину ускорения. Численное интегрирование ускорения позволяет перейти к скорости, а повторное интегрирование к перемещению. Таким образом, с учетом представленных выше замечаний и правилах перехода из физического пространства в географическое, появляется принципиальная возможность оценить геодезические координаты системы в любой момент времени.

### 1.1.2.5 Корректировка выходных данных

Так как процессы определения координат с использованием инерциальных устройств и визуальной одометрии проходят параллельно, то на выходе получают два значения для  $x$ , для  $y$  и для  $\alpha$ .

Практика показывает, что значения полученные при использовании визуальной одометрии более точные и величина ошибки меньше зависит от времени, в то время как инерциальные приборы подвержены «дрейфу нуля», о чем сказано в пункте 1.1.2.4.

Однако при передвижении по однотонной поверхности, на которой алгоритм выделения ключевых точек дает сбой, визуальная одометрия может давать совершенно непредсказуемые результаты. На практике это выражается в резком изменении выходных значений  $x$ ,  $y$ ,  $\alpha$ .

В связи с этим, данные, полученные с использованием инерциальных

устройств и визуальной одометрии, следует проверять на правдивость.

Следует выделить следующие контрольные величины, изменение которых следует учитывать.

### 1. Скорость перемещения.

Чаще всего известна максимальная скорость передвижения камеры, ограниченная максимальной скоростью перемещения ее носителя. Важно понимать, что  $x$ ,  $y$  и  $\alpha$  это изменения соответствующих параметров положения носителя в пространстве. Для перехода от них к скорости можно воспользоваться следующей формулой:

$$V \cdot \Delta t = x^2 + y^2$$

При обработке видеофайла можно узнать частоту кадров, которая чаще всего составляет 22 или 30 кадров в секунду. Тогда временной интервал между двумя кадрами, а следовательно и между двумя итерациями работы алгоритма визуальной одометрии составляет  $1/22 \approx 45\text{мс}$  и  $1/30 \approx 33\text{мс}$  соответственно. Будем обозначать этот интервал времени через  $\Delta t$ .

Таким образом можно установить следующий сигнальный показатель ошибки в работе алгоритма визуальной одометрии:

$$x^2 + y^2 \leq V_{\text{макс}} \cdot \Delta t = \frac{x^2 + y^2}{\Delta t} \leq V_{\text{макс}}$$

### 2. Абсолютная скорость изменения параметров.

Так как измерительные устройства расположены на каком-либо физическом носителе, и этот носитель обладает некоторой инерционностью, то скорость изменения параметров не должна быть выше, чем это позволяют динамические параметры носителя. Скорость изменения параметров есть ни что иное, как первая производная. Но если учитывать, что  $x$ ,  $y$  и  $\alpha$  это изменения соответствующих параметров положения носителя в пространстве, то необходимо ограничивать их величину на единицу времени.

Введем следующие сигнальные показатели:

- $x / \Delta t \leq V_{\text{макс}}^x = V_{\text{макс}}$
- $y / \Delta t \leq V_{\text{макс}}^y = V_{\text{макс}}$
- $\alpha / \Delta t \leq V_{\text{макс}}^\alpha$

### 3. Относительная скорость изменения параметров.

Так же как и в предыдущем пункте, инерционность носителя устройств накладывает ограничения на изменения параметров относительно предыдущей итерации. Пусть  $x_{\text{пред}}$  - значение, полученное на предыдущей итерации алгоритма визуальной одометрии, а  $x_{\text{тек}}$  - текущее значение. Таким образом  $x_{\text{тек}} - x_{\text{пред}}$  - разница в изменении положения носителя за два временных отрезка. С учетом физических ограничений, получаем:

$$x_{\text{тек}} - x_{\text{пред}} \leq \Delta X_{\text{max}}$$

Аналогично получаем для  $y$  и  $\alpha$ :

$$y_{\text{тек}} - y_{\text{пред}} \leq \Delta Y_{\text{max}}$$

$$\alpha_{\text{тек}} - \alpha_{\text{пред}} \leq \Delta \alpha_{\text{max}}$$

Для угла поворота следует учитывать то, что этот показатель может колебаться в пределах от  $0^\circ$  до  $360^\circ$ , и значение  $\alpha$  следует брать исключительно до аккумулялирования с общими значениями.

#### 1.1.2.6 Анализ функций, подлежащих автоматизации

Исходя из сказанного выше, можно выделить следующие функции, подлежащие автоматизации.

- Вычисление оптического потока.
- Обработка инерциальных данных и вычисление смещения
- Сопоставление полученных данных и выбор наилучшего результата.

### 1.1.3 Выбор критериев качества

Выделим основные критерии, по которым следует оценивать разрабатываемую подсистему.

- **Необходимость в специальном оборудовании** - исходя из задачи, предполагается использование проектируемого модуля в низкобюджетных системах. В связи с этим необходимо обеспечить корректную работу подсистемы с неспециальным оборудованием таким, как бытовые камеры.
- **Стоимость необходимого оборудования** – исходя из предыдущего пункта, следует обеспечивать поддержку наиболее дешевого оборудования.
- **Сложность интеграции** – в современном мире наличия многих конкурентов и налогов одним из ключевых факторов при выборе между ними является простота использования продукта. В связи с этим необходимо снизить время, необходимое на интеграцию с разрабатываемой подсистемой, путем предоставления удобного в использовании API.
- **Точность** - данный критерий является важным для задач определения положения в пространстве априори, так как при низкой точности использование систем данного рода становится бессмысленным. Тем не менее в рамках многих задач обеспечение чрезмерно высокой точности является избыточным.
- **Возможность свободного использования** – нередко разработчики программных или аппаратных продуктов накладывают ограничение в виде разного рода лицензий распространения, которые ограничивают пользователей в распространении и использовании своих продуктов. В связи с этим важным является свобода использования продукта в любых целях, не противоречащих законодательству стран, где происходит его использование.

## 1.1.4 Анализ аналогов и прототипов

### 1.1.4.1 ROS Odometry

ROS (Robot Operating System) — Операционная система для роботов — это фреймворк для программирования роботов, предоставляющий функциональность для распределённой работы. ROS был первоначально разработан в 2007 году под названием *switchyard* в Лаборатории Искусственного Интеллекта Стэнфордского Университета для проекта (STAIR). В 2008 году развитие продолжается в Willow Garage, научно-исследовательском институте/инкубаторе робототехники, совместно с более чем двадцатью сотрудничающими институтами.

ROS обеспечивает стандартные службы операционной системы, такие как: аппаратную абстракцию, низкоуровневый контроль устройств, реализацию часто используемых функций, передачу сообщений между процессами, и управление пакетами. ROS основан на архитектуре графов, где обработка данных происходит в узлах, которые могут получать и передавать сообщения между собой. Библиотека ориентирована на Unix-подобные системы (Ubuntu Linux включен в список «поддерживаемых» в то время как другие варианты, такие как Fedora и Mac OS X считаются «экспериментальными»).

ROS имеет две основные «стороны»: стороны операционной системы *ros*, как описано выше и *ros-pkg*, набор поддерживаемых пользователями пакетов (организованных в наборы, которые называются стек), которые реализуют различные функции робототехники: SLAM, планирование, восприятие, моделирование и др.[9]

В состав ROS так же входят различные пакеты одометрии, наиболее приемлемым является пакет LIBVISO2 [10].

LIBVISO2 (библиотека для визуальной одометрии, версия 2) является очень быстрой кроссплатформенной библиотекой, написанной на C++ с MATLAB оберткой для использования. Основная цель библиотеки – вычисление 6 DOF движения движущейся моно/стерео камеры. Не будем подробно рассматривать стереоверсию библиотеки, так как этот класс одометрии лежит в другой плоскости. Монокулярная версия на данный момент еще находится в статусе экспериментальной и использует 8-ми точечный алгоритм

для фундаментальной оценки матрицы. Кроме того, он предполагает, что камера движется при известной и фиксированной высоте над землей (для оценки масштаба).

Чтобы оценить масштаб движения, моноодометр использует горизонтальную плоскость и, следовательно, нуждается в информации о  $z$ -координате и углу наклона камеры.

В общем случае, монокулярная одометрия и SLAM-системы не могут оценить движение или положение на метрической шкале. Libviso2 решает эту неопределенность через фиксированный переход от плоскости земли к камере. Для этого и необходимо знать параметры *camera height* и *camera pitch* (высота камеры и ее угол наклона). Эти значения должны быть оценены и введены в систему вручную при каждой установке камеры на объект[11].

Кроме того, существует еще одна проблема: когда камера выполняет только чистое вращение, матрицы для решения СЛАУ вырождаются, и, даже если система очень мощная, алгоритм перестает корректно работать.

#### **1.1.4.2 Project Tango**

Небольшая группа инженеров АТАР (Advanced Technology and Projects) в компании Google занимается разработкой перспективных технологий. Сегодня она представила свой новый проект Tango. Это очень красивая технология построения 3D-модели окружающего пространства с помощью смартфона.

Группа АТАР сконструировала 5-дюймовый смартфон, оснащённый стереокамерой, сенсорами и программным обеспечением, которые отслеживают положение смартфона в 3D-пространстве, а также сканируют окружающий мир в реальном времени со скоростью 250 тыс. измерений в секунду. Всё это объединяется в единую 3D-модель с помощью уникального процессора Myriad 1, разработанного стартапом Movidius.

Когда модель готова, телефон может постоянно определять своё местоположение внутри неё. На базе этой модели можно создавать интересные игры, когда виртуальные объекты в смартфоне совмещаются с реальными объектами окружающей действительности. Например, вы можете играть на смартфоне в футбол, отбивая мячик от стены собственной кварти-

ры. Или управлять настоящим роботом в соседней комнате, двигая его по 3D-модели этой комнаты. Или играть в прятки с анимированным персонажем в своём собственном доме. В общем, возможности открываются невероятные.

В ближайшие месяцы ATAP намерены выпустить SDK для создания программ на новом оборудовании. Первые 200 устройств разошлют разработчикам после 14 марта. Чтобы принять участие в программе, нужно придумать интересное приложение в сфере навигации внутри помещений, игр для одного или нескольких игроков или алгоритмы обработки данных с сенсоров Tango.

Телефон работает под Android, так что API для доступа к информации об ориентации, позиционировании и расстоянии до объектов можно использовать со стандартными Android-приложениями, написанными на Java, C/C++, а также на игровом движке Unity. Первые версии алгоритмов и API скоро закончат, но проект и после этого останется в статусе эксперимента [12][13].

#### **1.1.4.3 SMP Robotics**

Компания SMP Robotics занимается созданием беспилотных транспортных средств и в их роботах серии «SRX» реализован принцип визуального определения местоположения робота.

Назначение роботов этой серии — езда по известному маршруту, особенности которого хранятся в бортовом компьютере робота в виде карты со множеством характеристик условий путей проездов. Более того, робот, многократно проезжая по одним и тем же участкам, может уточнять и накапливать данные о них на протяжении дня и ночи, сезонных или погодных изменениях окружающего ландшафта.

Всё это позволило успешно применить алгоритм визуального определения местоположения робота на основе анализа и последующего сравнения видеоизображений, полученных при первичном и последующем проездах. Первичный проезд можно рассматривать как обучающий, в нём формируется набор ключевых кадров, которые позволяют сформировать базу данных, описывающую взаиморасположение устойчивых структур. При последующем автоматическом движении характеристики текущего

изображения проверяются на тождественность описаний, хранящихся в базе данных, и, при их совпадении, осуществляется привязка к текущему местоположению.

В условиях циклических проездов робота по одному и тому же маршруту появляется возможность автоматического пополнения и уточнения базы данных взаиморасположения устойчивых структур на видео с курсовой камеры. Накопленные данные будут коррелировать со сменой дня и ночи, сезонными изменениями ландшафта, особыми погодными условиями. Расширенная база данных позволяет произвести достоверное сопоставление большому количеству текущих кадров видеоизображения к ранее обработанным и сохранённым кадрам известной траектории движения, тем самым достигнуть точного определения местоположения на большинстве участков траектории движения и, соответственно, повысить скорость автоматического проезда по известному маршруту.

Используемый метод визуального определения местоположения и управления движением позволяет переносить накопленную информацию о маршруте движения от одного робота к другому. Робот, проехавший по маршруту много раз, сформирует достаточно полную базу данных о его характеристиках. База данных может быть установлена на новый робот, который еще ни разу не проходил по данному маршруту, и он успешно проедет по нему с достаточной точностью и предельно возможной скоростью. Возможность копирования и обмена данными о маршруте, накопленными алгоритмом визуального определения местоположения, особенно полезна при проездах по значительным территориям. На начальном этапе несколько роботов накапливают информацию на локальных участках, при достижении её достаточного качества, все данные объединяются в единую базу данных и представляют собой информацию о маршруте в целом. Объём этих данных позволяет каждому из роботов проехать все участки маршрута, не смотря на то, что конкретно этот робот там никогда не ездил. Информация о маршруте длительное время остается достоверной. Несмотря на естественные изменения ландшафта, адаптивность алгоритма позволяет скорректировать массив данных даже при значительных изменениях окружающей обстановки, а при последующих проездах снова пополнить его новыми данными.



Метод визуального определения местоположения позволяет обойтись без внешних, по отношению к движущемуся роботу, компонентов системы. Информации, полученной в результате обработки изображения окружающего ландшафта, достаточно для достоверного принятия решения о выборе пути движения. И это — коренное отличие описываемого подхода по сравнению с известным методом высокоточного вождения по данным от дифференциального приемника спутниковой навигационной системы. Такая система требует не только прямой видимости на спутниковую группировку, но и базовой корректирующей станции в зоне радиовидимости робота. В реальных условиях эксплуатации каналы связи, как со спутниками, так и с базовой станцией нельзя считать надежным. Даже в городском лесопарке с низкой плотностью деревьев получить достоверные данные от спутниковой навигационной системы практически невозможно, кроме того, существуют зоны радио тени, обусловленные застройкой. Высокочастотный спутниковый сигнал подвержен серьезному ослаблению даже во время незначительных осадков, мокрая листва деревьев вносит достаточное ослабление, чтобы сделать спутниковую группировку радио невидимой. В целом, метод высокоточного вождения по данным от дифференциального приемника спутниковой навигационной системы, хорош в условиях, для которых он и разрабатывался — для автоматического вождения сельскохозяйственной техники в чистом поле и, желательно, без дождя.

Управление движением посредством анализа видеоизображений окружающей обстановки и визуального определения местоположения обеспечивает полную автономность движения наземного робота. И, в отличие от других рассмотренных методов, не требует ни внешней инфраструктуры, ни устойчивых каналов связи с дополнительным оборудованием [14].

#### **1.1.4.4 Сравнительный анализ**

Для сравнения представленных вариантов воспользуемся методом взвешенной суммы. Данный метод позволяет объединить ряд критериев сравнения в один интегральный показатель, по которому затем выбирается наилучший вариант, соответствующий максимальному значению этого интегрального показателя. Метод взвешенной суммы можно представить

следующим образом:

$$Y = \max_{j \in m} \sum_{i=1}^n \alpha_i \cdot K_{ij},$$

где  $\sum_{i=1}^n \alpha_i = 1$

По этому критерию проводится сравнение  $j(j = 1, 2, \dots, m)$  вариантов по  $i(i = 1, 2, \dots, n)$  показателям, где:

$n$  – количество показателей сравнения;

$m$  – количество вариантов сравнения.

$K_{ij}$  – нормированный коэффициент соответствия  $i$ -ого параметра  $j$ -ого варианта эталонному значению, т.е. для  $j$ -ого варианта:

$$K_{ij} = \frac{\max_j X_{ij}}{X_{ij}},$$

$$0 < K_{ij} < 1$$

Соответствие систем-аналогов выбранным критериям качества представлено в Таблице 1.

Таблица 1: Сравнение аналогов. Лингвистические оценки.

Критерий	Проектируемая подсистема	Аналоги		
		ROS odometry	Project tango	SMP Robotics
Необходимость в специальном оборудовании.	Нет	Нет	Да	Да
Стоимость необходимого оборудования	Низкая	Средняя	Средняя	Высокая
Сложность интеграции.	Низкая	Высокая	Низкая	Средняя
Точность.	Средняя	Высокая	Высокая	Высокая
Возможность свободного использования.	Есть	Есть	Есть	Нет

Для сравнений необходимо перевести качественные оценки в коли-

ественные. Метод перевода для каждого из критериев представлен в таблицах ниже.

Таблица 2: Сравнение аналогов. Лингвистические оценки.

Описание	Для функционирования системы необходимы специфические устройства, или устройства с повышенными точностями	Система может функционировать на любом оборудовании и это оборудование не сложно найти
Лингв. оценка	Да	Нет
Балл	0,4	1

Таблица 3: Перевод качественных оценок в количественные для критерия «Стоимость необходимого оборудования»

Описание	Стоимость необходимого оборудования исчисляется десятками тысяч рублей	Стоимость оборудования лежи в диапазоне от 5 000 до 20 000 рублей	Стоимость оборудования составляет менее 5 000 рублей
Лингв. оценка	Высокая	Средняя	Низкая
Балл	0,1	0,6	1

Учтём весовой коэффициент. Для этого каждому критерию зададим весовой коэффициент, кратный  $x$  - минимальному весовому коэффициенту:

1. Необходимость в специальном оборудовании –  $3x$
2. Стоимость необходимого оборудования –  $3x$
3. Сложность интеграции –  $2x$
4. Точность –  $2x$
5. Возможность свободного использования –  $1x$

Таблица 4: Перевод качественных оценок в количественные для критерия «Сложность интеграции»

Описание	Данную систему сложно интегрировать с другими системами, требуются серьезные доработки	Данную систему можно интегрировать с другими системами, но для этого необходимы доработки обеих систем	Данную систему легко интегрировать с другими системами, стоимость и время доработок низкие
Лингв. оценка	Высокая	Средняя	Низкая
Балл	0,1	0,6	1

Таблица 5: Перевод качественных оценок в количественные для критерия «Точность»

Описание	Точность ограничена лишь точностью используемого оборудования, со временем увеличение ошибки не наблюдается.	Используемые алгоритмы допускают определенную погрешность, со временем возможно увеличение ошибки	Точность вычислений крайне низкая, со временем наблюдается накопление ошибки
Лингв. оценка	Высокая	Средняя	Низкая
Балл	1	0,6	0,1

Таблица 6: Перевод качественных оценок в количественные для критерия «Возможность свободного использования»

Описание	Исходные коды открыты, допускается свободное использование	Система является проприетарной
Лингв. оценка	Да	Нет
Балл	1	0,4

Сумма весовых коэффициентов по всем критериям качества равна 1.

$$3x + 3x + 2x + 2x + 1x = 11x \Rightarrow x = 0,0909$$

Таблица 7: Сравнение аналогов. Количественные оценки, весовой коэффициент.

Критерий	Вес. коэф.	Проектируемая подсистема	Аналоги		
			ROS odometry	Project tango	SMP Robotics
Необходимость в специальном оборудовании.	0,2727	1	1	0,4	0,4
Стоимость необходимого оборудования	0,2727	1	1	0,4	0,4
Сложность интеграции.	0,2727	1	1	0,4	0,4
Точность.	0,2727	1	1	0,4	0,4
Возможность свободного использования.	0,2727	1	1	0,4	0,4

Таким образом, можно составить сравнительную таблицу, содержащую количественные и качественные оценки и подсчитать итог, метом взвешенной суммы. Итоговое сравнение с учетом весового коэффициента и количественных оценок приведены в Таблице 7.

#### 1.1.5 Перечень задач, подлежащих решению в процессе разработки

Исходя из приведенного выше первичного анализа предметной области можно сформировать список задач, подлежащих решению.

Необходимо решить следующие задачи:

1. разработка структуры и архитектуры подсистемы системы;
2. разработка требований к формату и структуре передаваемых данных;
3. разработка алгоритмов обработки информации;
4. выбор и обоснование КТС, необходимого для реализации системы;
5. оценка предполагаемого качества функционирования системы;

6. организационно-экономическое обоснование разработки;
7. рекомендации по охране труда.

## **1.2 Проектирование подсистемы**

### **1.2.1 Разработка структуры подсистемы**

#### **1.2.1.1 Определение состава компонентов**

Исходя из анализа функций структурно в подсистеме можно выделить следующие основные части:

- **модуль обработки входных данных** (преобразует входные данные в удобоваримый вариант для последующей обработки);
- **модуль компьютерного зрения** (позволяет обрабатывать изображения и производить их анализ для построения визуальной одометрии);
- **модуль визуальной одометрии** (высчитывает перемещение и угол поворота камеры на основе последовательности изображений);
- **модуль обработки данных с инерционных приборов** (производит математическую обработку показаний датчиков и на ее основе вычисляет перемещение объекта);
- **модуль сопоставления и вывода данных** (сравнивает показания двух предыдущих модулей и на их основе выводит наиболее правдоподобное положение объекта).

#### **1.2.1.2 Определение структуры компонентов**

Согласно общепринятой терминологии система состоит из подсистем, а те в свою очередь из модулей. Таким образом разрабатываемая подсистема состоит из следующих модулей:

- обработки входных данных;
- обработки выходных данных;

- визуальной одометрии;
- компьютерного зрения;
- настроек;
- одометрии на основе инерционных устройств.

Графически архитектура подсистемы представлена на рисунке 4.



Рис. 4: Структура компонентов.

### Назначение модулей

- **Модуль обработки входных данных** - нормализует входные данные и аккумулирует их, в случае слишком высокой частоты их поступления.
- **Модуль обработки выходных данных** - сравнивает данные, полученные независимо в модуле визуальной одометрии и в модуле обработки данных с ИИУ, и принимает решение об их корректности на основе определенных сигнальных показателей (см. пункт 1.1.2.5).
- **Модуль визуальной одометрии** – вычисляет перемещения на основе видеоряда. Для обработки видео использует модуль компьютерного зрения.
- **Модуль компьютерного зрения** – реализует стандартные алгоритмы компьютерного зрения, такие как вычисление оптического потока, поиск опорных точек на изображении, трансформация и изменение изображений.

- **Модуль одометрии на основе инерционных устройств** – производит расчеты перемещения объекта на основе полученных данных с гироскопов и акселерометров.
- **Модуль настроек** - содержит в себе данные об используемом оборудовании и его характеристиках. Эти данные используются в модулях одометрии.

### 1.2.1.3 Описание процессов

В ходе работы всей подсистемы протекает множество процессов по обработке и преобразованию информации. Выделим ключевые из них.

#### **Визуальная одометрия.**

В данной работе делается упор на создание визуальной одометрии на основе одной бытовой камеры и относительно слабых вычислительных устройств. Такое решение требует соблюдения двух условий, которые приемлемы при создании большинства современных мобильных роботов:

1. перемещение происходит по земле или другой горизонтальной плоскости;
2. камера жестко закреплена относительно самого носителя.

При соблюдении этих условий работу визуальной одометрии можно описать в 8 шагов[15].

1. Исправление изображения для исключения искажения линз.
2. Вычисление оптического потока для кадра.
3. Проверка полученных векторов смещения, исключение движущихся в кадре объектов, исключение ошибочных векторов.
4. Разделение всего оптического потока на две части: «наземную» и «небесную».
5. В «небесной» части перейти к цилиндрической системе координат и вычислить угол поворота относительно двух последних кадров, определяя тем самым угол поворота  $\theta$ .



6. В «наземной» части выделить векторы  $(u, v)$  из оптического потока и вычислить перемещение в плоскости  $x-y$ , получив вектор  $(x, y)$ .
7. Прибавить  $(x, y, \theta)$  к изначальному положению объекта  $(X, Y, \Theta)$ , получив новое положение объекта.
8. Перейти к шагу 1 для следующего кадра. Периодически обновлять ключевые точки.

В общем виде процесс вычисления оптического потока представлен на рисунке 5.

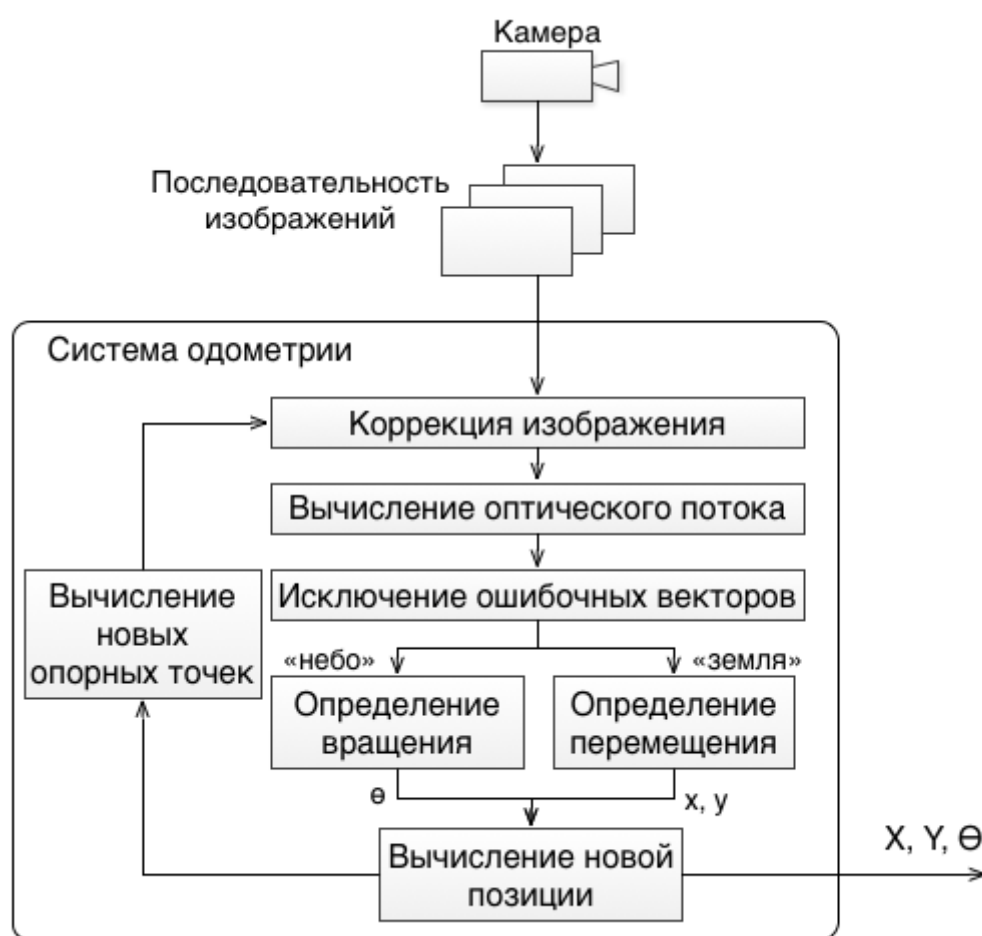


Рис. 5: Графическое представление процесса визуальной одометрии

### **Обработка данных с инерционных измерительных устройств.**

Обработка данных с инерционных устройств представляет собой несколько операций интегрирования полученных данных. Рассмотрим переход от ускорений к перемещению.

Пусть на выходе акселерометра снимаются ускорения  $Z_a, Y_a$  и  $X_a$ , которые показывают ускорения по осям  $X, Y, Z$ .

Как мы знаем из курса физики,  $\int_a^b a(t)dt = V(t) + C$ . В то же время  $\int_a^b V(t)dt = S(t) + C$ . Таким образом получаем:

$$\int \int_a^b a(t)dt = \int_a^b V(t) + C_v = S(t) + C_v \cdot t + C_s$$

.

Так как  $C_s$  в нашем случае, это начальное положение носителя, то его мы принимаем  $= 0$ . Отсюда получаем формулу:

$$\int \int_a^b a(t)dt = \int_a^b V(t) + C_v = S(t) + C_v \cdot t.$$

Однако все упрощается, если период замеров настолько мал, что ускорение можно считать равномерным. Тогда формулы преобразуют следующий вид:

$$V_t = V_{t-1} + a_t \cdot t$$

$$S_t = S_{t-1} + V_t \cdot t$$

Графически это можно выразить тремя графиками (рис. 6): показывающими рассчитанные значения перемещения и скорости в зависимости от времени.

Производя подобные математические вычисления для координат  $X$  и  $Y$  получаем перемещения по этим координатам. Причем мы можем опускать финальную константу интегрирования, так как нам необходимо определить не пройденный путь от начала движения, а именно путь, пройденный за последний временной отрезок.

Аналогично обрабатываются данные с акселерометров, с одним лишь отличием - акселерометры могут предоставлять сразу угловую скорость, что позволяет производить лишь одно интегрирование.

#### 1.2.1.4 Математическое обеспечение

В рамках данной работы многие процессы основываются на фундаментальных исследованиях в области компьютерного зрения и обработки

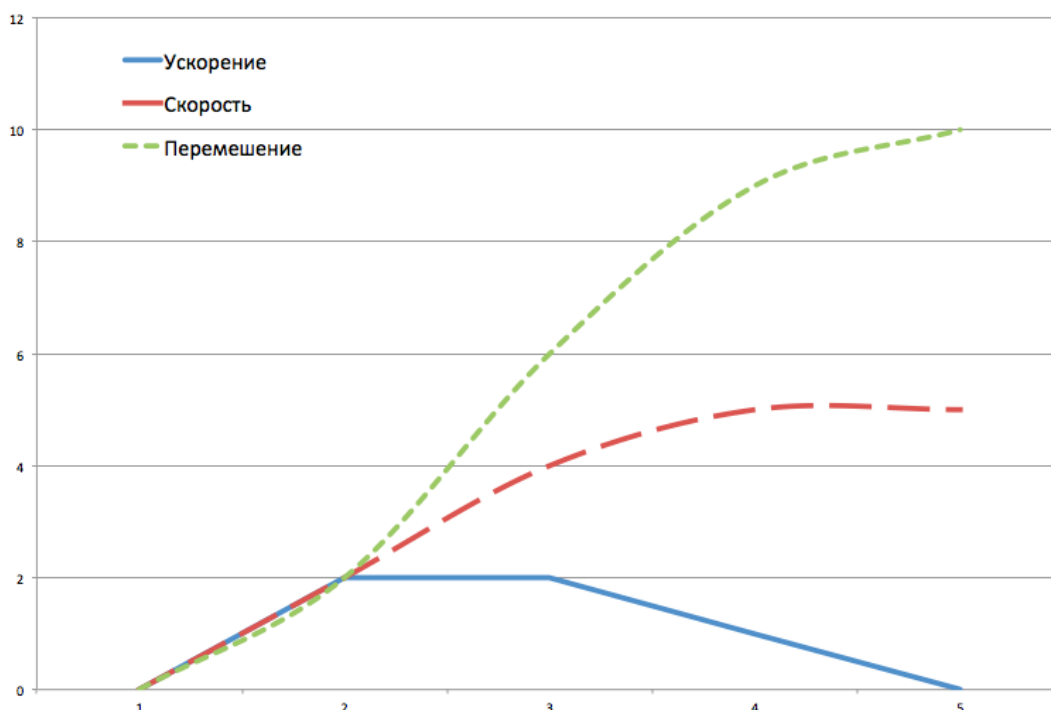


Рис. 6: Пример интегрирования ускорения для получения перемещения

изображений. Далее рассматриваются основные из них.

### Поиск ключевых точек

Есть много видов местных ключевых точек, которые можно отслеживать. Для начала стоит рассмотреть что из себя представляет ключевая точка сама по себе (ключевая особенность изображения). Очевидно, что если мы выбираем точку на большой однотонной стене, то будет не легко найти, что же точку в следующем кадре из видео.

Если все точки на стене могут быть одинаковыми или даже очень похожи, то у нас будет мало шансов отслеживания этой точки в последующих кадрах. С другой стороны, если мы выберем точку, которая является уникальной, то у нас будут довольно хорошие шансы снова найти эту точку. На практике, точка или черта, которая мы выбираем в качестве ключевой должны быть уникальным или почти уникальными, и должна быть параметризуемой таким образом, чтобы ее можно было отличить от других точек на изображении.

Возвращаясь к нашей аналогии с большой пустой стеной, мы могли бы попытаться искать точки, которые имеют некоторые значительные изменения в окрестности, например, большую производную. На практике оказывается, что этого не достаточно. Точка, в которой большое значение

производной, может находиться на какой-то линии, но все точки на этой линии будут иметь такую же или близкую производную.

Однако, если высокие значения производные наблюдаются в двух ортогональных направлениях, то можно надеяться, что эта точка будет уникальной. По этой причине, такие особенности на изображении называются углами. Очевидно, углы - не края - это точки, которые содержат достаточно информации, чтобы ее нашли на последующих кадрах.

Определение углов опирается на матрице производных второго порядка  $(\partial^2 x, \partial^2 y, \partial x \partial y)$  интенсивностей изображения. Эта терминология происходит от матрицы Гессе вокруг точки, которая определяется в двух измерениях следующим образом:

$$H(p) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}$$

Такие углы, по определению Харриса, места на изображении, где автокорреляционная матрица вторых производных имеет два больших собственных значения. По сути, это означает, что есть текстуры или грани, идущие, по крайней мере, в двух отдельных направлениях, сосредоточенные вокруг такой точки, так же, как реальные углы имеют по крайней мере два ребра, сходящихся в точку. Использование вторых производных позволяет точно определить особенности, потому что они не отвечают единым градиентами (так как при первой производной равной константе, вторая будет равна нулю). Это определение имеет дополнительное преимущество, в том, что когда мы рассматриваем только собственных значений автокорреляционной матрицы, мы рассматриваем величины, инвариантные также к вращению, что является важным, потому что объекты, которые мы отслеживаем могут вращаться, а также перемещаться. Следует также отметить, что эти два собственных значения делают больше, чем просто определяют, является ли точка перспективной для отслеживания - они также обеспечивают идентифицирующую роль для точки.

В используемом в данной работе пакете компьютерного зрения используется функция *cvGoodFeaturesToTrack()*.

```
void cvGoodFeaturesToTrack(
```

```

    const CvArr* image,
    CvArr* eigImage, CvArr* tempImage,
    CvPoint2D32f* corners,
    int* cornerCount,
    double qualityLevel,
    double minDistance,
    const CvArr* mask=NULL,
    int blockSize=3,
    int useHarris=0,
    double k=0.04
);

```

Эта функция вычисляет вторые производные для точек и их собственные значения. Результатом работы функции является список точек, которые являются хорошими кандидатами для ключевых точек.

### **Оптический поток**

Задача вычисления оптического потока можно сформулировать следующим образом: оценить движение между двумя кадрами не имея никакой информации о происходящем, кроме самих кадров[16].

Вычисляя оптический поток, мы можем связать для каждого пикселя определить некий вектор смещения, которые будут показывать куда переместился пиксель между предыдущим кадром и текущим кадром. Такой подход обычно называют плотным оптическим потоком, который определяет перемещение каждого пикселя. Метод Хорн-Шунка (Horn-Schunck) вычисляет именно такой оптический поток. В его основе лежит один, казалось бы, простой принцип - просто пытаться найти наиболее похожий пиксель на следующем кадре в пределах какого-то окна вокруг исходного пикселя.

На практике расчет плотного оптического потока затруднителен. Рассмотрим движение белого листа бумаги. Многие из белых пикселей в предыдущем кадре просто остаются белыми на следующих. Изменения будут лишь на границах листа, и то, только вокруг границ перпендикулярных движению. В результате необходимо прибегать к различным математическим приемам, что сказывается на увеличении ресурсоемкости операции в целом.

Это приводит нас к альтернативному варианту, выборочный опти-

ческого потока. Алгоритмы такого рода опираются на некоторые средства определения заранее подмножество точек, которые должны быть отслежены. Если эти точки имеют определенные свойства, такие как свойства ключевых особенностей, обсуждаемые ранее, тогда отслеживая будет относительно точным и надежными. Для многих практических случаев, вычислительная стоимость выборочного оптического потока намного меньше, чем у плотного оптического потока, поэтому последнему отводится только академический интерес.

Рассмотрим наиболее популярный метод вычисления выборочного оптического потока - метод Лукаса-Канаде (Lucas-Kanade). Этот метод также имеет реализацию, которая работает с пирамидами изображений, что позволяет нам отслеживать быстрые движения. В данной работе используется именно он, так как он обладает наиболее низкой вычислительной сложностью.

### ***Метод Лукаса-Канаде***

Метод (алгоритм) Лукаса - Канаде (ЛК) создавался в 1981 году и первоначально задумывался для вычисления плотного оптического потока. Тем не менее, Алгоритм работал и с любым количеством точек для отслеживания, что позволило использовать его в столь важных выборочных оптических потоках.

Алгоритм ЛК может быть применен для определенного числа точек, потому что он опирается только на локальной информации о точке, которая является производной в некотором маленьком окне вокруг каждой из ключевых точек. Недостатком использования небольших локальных окон в ЛК является то, что большие смещения могут перемещать точки за пределы таких локального окон, что приведет к невозможности их нахождения. Эта проблема привела к разработке "пирамидальной" версии алгоритма, которая составляет пирамиду из нескольких копий изображения разного размера, после чего вычисляет оптический поток, начиная с самого высокого уровня в пирамиде изображений, постепенно опускаясь по уровням для более высокой точности.

### **Принцип работы алгоритма**

Основная идея алгоритма ЛК основывается на трех предположениях.

1. *Яркость постоянна.* Предполагается, что пиксель не меняет внешний

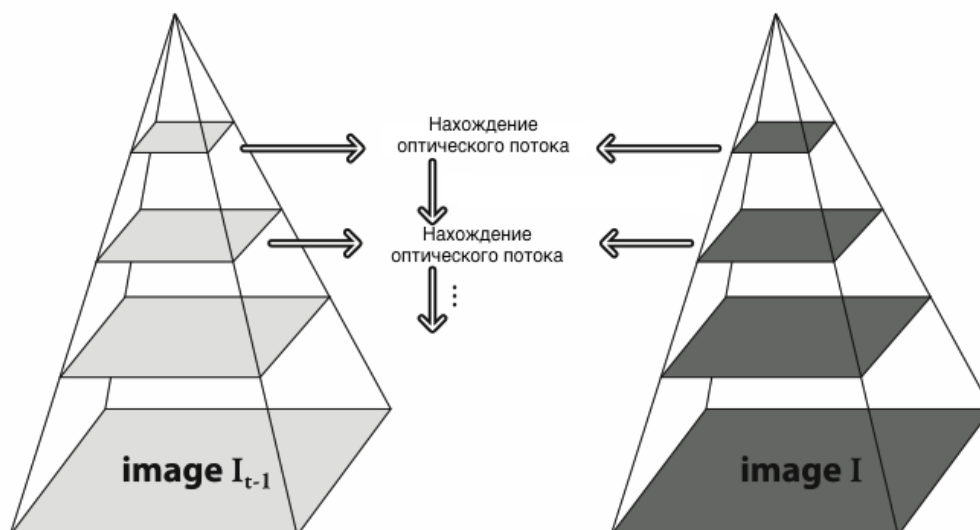


Рис. 7: Графическое представление работы "пирамидальной" версии алгоритма ЛК

вид, переходя от кадра к кадру. Для изображений в градациях серого (для случая с цветными изображениями есть более строгое допущение, но оно сводится к этому) это означает, что яркость пикселя не меняется, при его отслеживании от кадра к кадру.

2. *Малые сдвиги.* Изображение движется медленно во времени. На практике это означает, что объект от кадра к кадру сдвигается незанчительно.
3. *Пространственная когерентность.* Соседние точки в кадре принадлежат к одной и той же поверхности и имеют аналогичные смещения.

Из первого допущения следует:

$$f(x, t) \equiv I(x(t), t) = I(x(t + dt), t + dt)$$

Отсюда следует, что интенсивность отслеживаемых пикселей не изменяется с течением времени:

$$\frac{\partial f(x)}{\partial t} = 0$$

Второе предположение, по существу, означает, что движения от кадра к кадру крайне малы. Другими словами, мы можем рассматривать это

изменение как аппроксимацию производной от интенсивности по времени. Чтобы понять последствия этого предположения, рассмотрим сначала случай одного пространственного измерения. Запишем уравнение яркости  $F(x, t)$  с учетом зависимости  $x$  от  $t$  и применим правило частного дифференцирования:

$$\underbrace{\frac{\partial I}{\partial x}}_{I_x} \bigg|_t \underbrace{\left( \frac{\partial x}{\partial t} \right)}_V + \underbrace{\frac{\partial I}{\partial t}}_{I_t} \bigg|_{x(t)} = 0,$$

где  $I_x$  является пространственной производной по первому изображению,  $I_t$  это производная между изображениями в течение долгого времени, и  $V$  - скорость, которую мы ищем. Таким образом, мы приходим к простому уравнению для скорости оптического потока в простой одномерной случае:

$$V = \frac{I_t}{I_x}$$

Рассмотрим стандартную задачу в одномерном случае. На рисунке 8 представлена ее графическая иллюстрация. Кривая  $I(x, t)$  изображает некую грань, слева от которой значения интенсивности высоки, а справа - низкие. Необходимо определить как сдвинулась эта грань на следующем кадре.

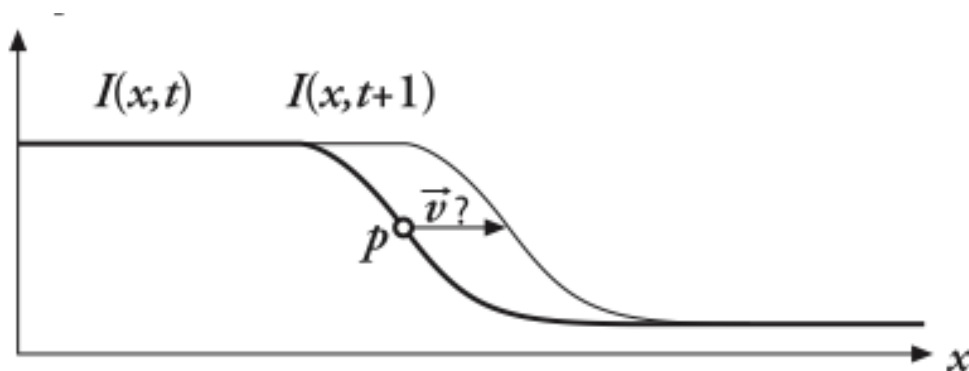


Рис. 8: Графическое представление задачи нахождения оптического потока в одномерном случае.

На рисунке 9 показано как можно решить такую задачу. При учете двух первых предположений получим следующее:

$$I_x = \frac{\partial I}{\partial x} \bigg|_t \text{ и } I_t = \frac{\partial I}{\partial t} \bigg|_{x=p} \Rightarrow \vec{V} \approx -\frac{I_t}{I_x}$$



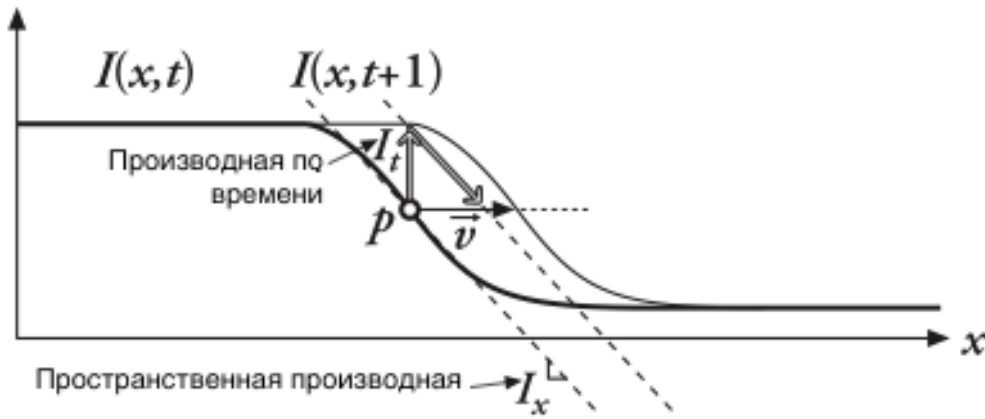


Рис. 9: Графическое представление решения задачи нахождения оптического потока в одномерном случае.

Теперь рассмотрим двумерный случай. Будем обозначать  $u_y$  скорость вдоль оси  $Y$ , а  $u_x$  -вдоль  $X$ :

$$I_x u + I_y v + I_t = 0 \equiv \frac{\partial I}{\partial t} + u_x \cdot \frac{\partial I}{\partial x} + u_y \cdot \frac{\partial I}{\partial y} = 0$$

Полученное уравнение, говорит нам о том, что сумма частных производных должны быть равна нулю. Но возникает проблема - уравнение у нас одно, а неизвестных в нем два:  $u_x$  и  $u_y$ .

Воспользовавшись третьим предположением, о том, что соседние пиксели смещаются на одинаковое расстояние, запишем это же уравнение для окна 5x5 пикселей, получив 25 уравнений. Очевидно, что 3 допущение не всегда верно, поэтому в общем случае система не имеет решения поэтому перейдем к минимизации ошибки:

$$E(u_x, u_y) = \sum_{i,j} g(x_i, y_i) \left[ \frac{\partial I}{\partial t} + u_x \cdot \frac{\partial I}{\partial x} + u_y \cdot \frac{\partial I}{\partial y} \right]^2$$

Здесь  $g$  — это функция, определяющая весовые коэффициенты для пикселей. Самый распространенный вариант — двумерная гауссиана, которая дает наибольший вес центральному пикселю и все меньший по мере удаления от центра (см. рис. 10).

Чтобы найти минимум  $E(u_x, u_y)$  воспользуемся методом наименьших квадратов, найдем её частные производные по  $u_x$  и  $u_y$  и запишем в более

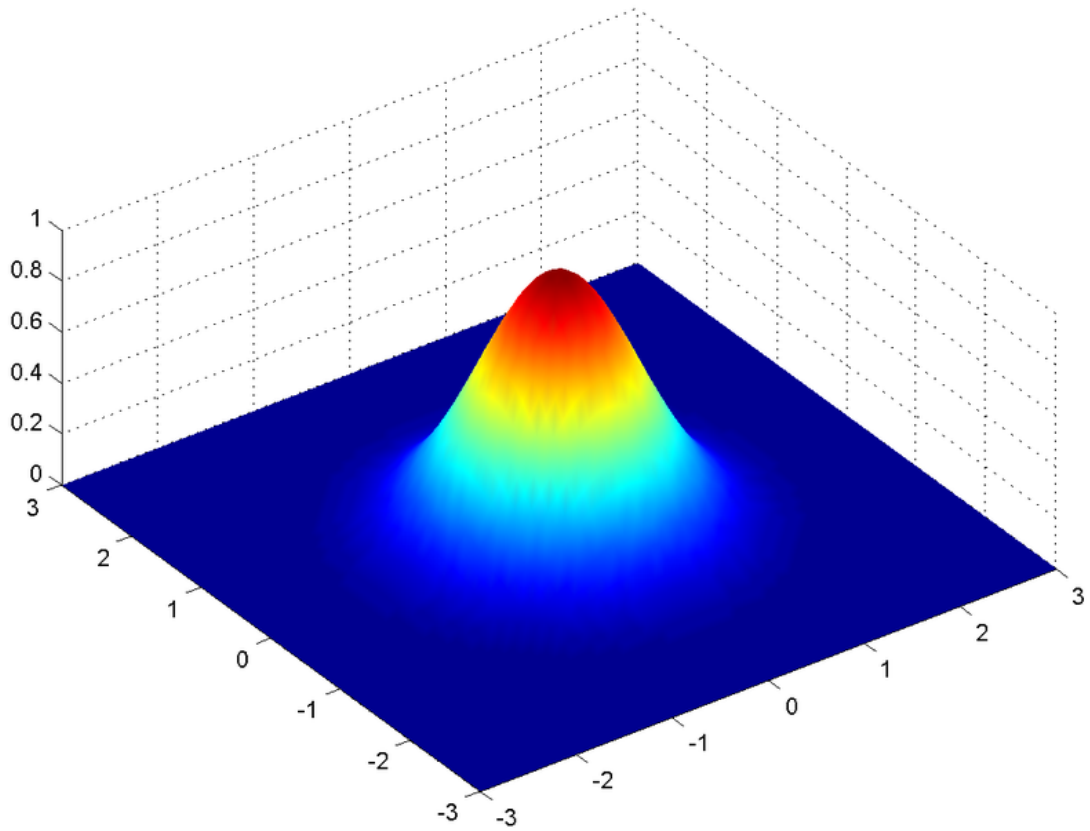


Рис. 10: Двумерная гауссиана

компактной форме, приравняем к 0:

$$\frac{\partial E(u_x, u_y)}{\partial u_x} = \sum_{i,j} g(x_i, y_i) \left[ u_x \left( \frac{\partial I}{\partial x} \right)^2 + u_y \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} + \frac{\partial I}{\partial x} \frac{\partial I}{\partial t} \right] = 0$$

$$\frac{\partial E(u_x, u_y)}{\partial u_y} = \sum_{i,j} g(x_i, y_i) \left[ u_x \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} + u_y \left( \frac{\partial I}{\partial y} \right)^2 + \frac{\partial I}{\partial x} \frac{\partial I}{\partial t} \right] = 0$$

Перепишем эти два уравнения в матричной форме:

$$M\vec{u} = \vec{b}$$

Где:

$$M = \begin{bmatrix} \sum_{i,j} g(x_i, y_i) \left( \frac{\partial I}{\partial x} \right)^2 & \sum_{i,j} g(x_i, y_i) \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \sum_{i,j} g(x_i, y_i) \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \sum_{i,j} g(x_i, y_i) \left( \frac{\partial I}{\partial y} \right)^2 \end{bmatrix}$$

$$\vec{b} = - \begin{bmatrix} \sum_{i,j} g(x_i, y_i) \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \\ \sum_{i,j} g(x_i, y_i) \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \end{bmatrix}$$

$$\vec{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

Если матрица  $M$  обратима (имеет ранг 2), можем вычислить  $u_x$  и  $u_y$ , которые минимизируют ошибку  $E[3]$ :

$$\hat{u} = M^{-1} \cdot \vec{b}$$

**Калибровка камеры** Еще одним математическим аспектом данной работы является калибровка камеры.

**Калибровка камеры** — это задача получения внутренних и внешних параметров камеры по имеющимся фотографиям или видео, отснятым ей[?].

Рассмотрим простейшую модель камеры, модель камеры с тонкой линзой. В этой простой модели, свет рассматривается в качестве потка от сцены или удаленного объекта, но только один луч попадает из любой конкретной точки этого объекта. Все эти точки проецируются на матрицу камеры или другую поверхность изображения. В результате, изображение на этой плоскости изображения всегда в фокусе, а масштаб изображения относительной его реального размера определяется одним параметром камеры - его фокусным расстоянием. На рисунке 11 схематично показана рассматриваемая модель камеры[?].

Из изображения видно, что, на основе подобия треугольников,  $-x/f = X/Z$ :

$$-x = f \cdot \frac{X}{Z}$$

Учитывая тот факт, что главная оптическая ось пересекает плоскость изображения не в координате  $(0, 0)$ , в координате  $c$  получим:

$$-x = f \cdot \frac{X}{Z} + c \quad (1)$$

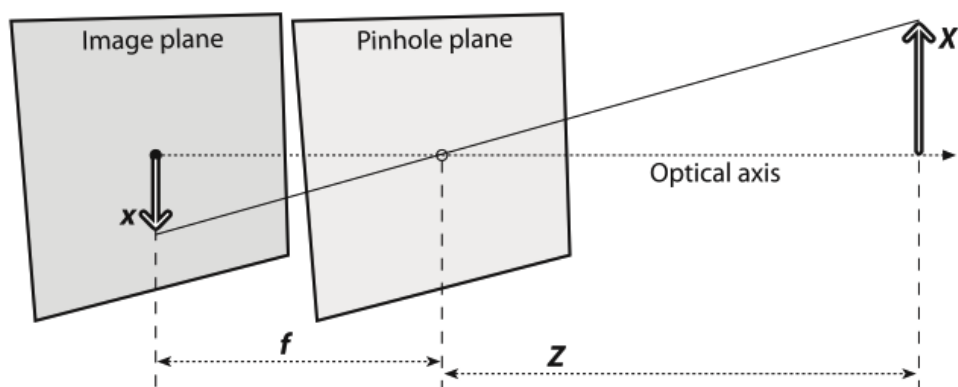


Рис. 11: Модель камеры с тонкой линзой

Далее будем рассматривать именно цифровые камеры с ПЗС-матрицами. У данных камер есть одна особенность - пиксели матрицы не квадратной формы из-за технологических условий изготовления. Учитывая этот факт, распишем уравнение 1 для координат  $x$  и  $y$ :

$$u = f \cdot s_u \cdot \frac{X}{Z} + c_u$$

$$v = f \cdot s_v \cdot \frac{X}{Z} + c_v$$

где  $s_u$  и  $s_v$  - коэффициенты формы пикселя.

### 1.2.2 Разработка формата и структуры данных

В ходе функционирования подсистемы информация поступает на ее вход, получается на ее выходе, и передается между модулями внутри системы. При этом эта информация носит разные характер и смысл. Так как подсистема носит сугубо программный характер, то все потоки данных реализуются только программными средствами, в виде структур данных. Сетевые технологии не накладывают никаких ограничений на форматы информационных структур. Ниже дано описание основных форматов обмена информацией.

- **Видео (вход).** Видео подается на вход системы покадрово, в связи с этим алгоритм кодирования и сжатия видео не существенен, так как задача сформировать следующий кадр в его полном объеме ложиться на вызывающую сторону. По сути, каждый кадр представляет собой 1

картинку в формате JPEG, что проще всего представлять матрицей, размер которой соответствует размеру кадра, а ее элементами являются структуры данных, хранящие информацию о каналах изображения.

- **Ускорения по трем осям (вход).** Данные подаваемые с акселерометров или одного, трехосевого акселерометра. Данные имеют цифровое представления, дробное.
- **Угловое ускорение по трем осям (вход).** Данные подаваемые с гироскопов или одного, трехосевого гироскопа. Данные имеют цифровое представления, дробное.
- **Текущие координаты и угол поворота (выход).** Высчитанное на текущей итерации положение объекта. Две координаты  $X$  и  $Y$  представляют собой целые числа. Угол поворота  $\alpha$  - дробное число. Аналогичным образом можно представлять информацию о смещениях, полученных на текущей итерации.
- **Информационный обмен между модулем визуальной одометрии и модулем компьютерного зрения.** Данный информационных обмен происходит два раза за итерацию: для нахождения ключевых точек на кадре и для вычисления оптического потока. При этом в разных случаях содержание информационных сообщений будет разное.
  - Нахождение ключевых точек - от модуля визуальной одометрии поступает изображение, являющееся кадром. Представляет собой матрицу, размер которой соответствует размеру кадра, а ее элементами являются структуры данных, хранящие информацию о каналах изображения. В ответ модуль компьютерного зрения возвращает список точек в формате  $x, y$ , где  $x$  и  $y$  - позиция пикселя на изображении при начале отсчета в левом верхнем углу.
  - Вычисление оптического потока - от модуля визуальной одометрии поступает два изображения (текущий и предыдущий кадры) и список ключевых точек. Представление соответствует описанному ранее. В ответ модуль компьютерного зрения возвращает два списка: список смещений, представляющий собой список пар  $\Delta x, \Delta y$ , и вектор ошибок, состоящий из 0 и 1 - 1 означает, что соответствующая ключевая

точка не была найдена на втором изображении и вектор смещения найден ошибочно или не найден вообще.

### **1.2.3 Разработка алгоритмов обработки информации**

#### **1.2.3.1 Общий алгоритм работы**

#### **1.2.3.2 Алгоритм вычисления оптического потока**

#### **1.2.3.3 Алгоритм обработки данных с ИИУ**

#### **1.2.3.4 Алгоритм сопоставления и корректировки выходных данных**

## Список литературы

1. Р.В. Жуков. Методы оценки положения объекта в пространстве // Наука и образование: научное издание МГТУ им. Н.Э. Баумана Электрон. журн. 2012. № 10. 2012. Режим доступа: <http://sntbul.bmstu.ru/doc/636938.html> [Дата обращения 05.11.2013].
2. Оптический поток. Свободная энциклопедия Википедия. Режим доступа: [http://en.wikipedia.org/wiki/Optical\\_flow](http://en.wikipedia.org/wiki/Optical_flow). [Дата обращения 10.03.2014].
3. То, что вы хотели знать про оптический поток, но стеснялись спросить. Коллективный блог Habrahabr. Режим доступа: <http://habrahabr.ru/post/201406/>. [Дата обращения 14.03.2014].
4. Вычисление оптического потока методом Лукаса-Канаде. Теория. Коллективный блог Habrahabr. Режим доступа: <http://habrahabr.ru/post/169055/>. [Дата обращения 14.03.2014].
5. Медведев Е.М. Данилин И.М. Мельников С.Р. ЛАЗЕРНАЯ ЛОКАЦИЯ ЗЕМЛИ И ЛЕСА [Электронный ресурс] // электронная версия книги. Режим доступа: <http://www.laserlocation.ru/chapter4.4.htm>. [Дата обращения 21.10.2013].
6. Рыскание. Свободная энциклопедия Википедия. Режим доступа: <http://ru.wikipedia.org/wiki/%D0%A0%D1%8B%D1%81%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5>. [Дата обращения 10.05.2014].
7. Крен. Свободная энциклопедия Википедия. Режим доступа: [http://en.wikipedia.org/wiki/Angle\\_of\\_list](http://en.wikipedia.org/wiki/Angle_of_list). [Дата обращения 10.05.2014].
8. Тангаж. Свободная энциклопедия Википедия. Режим доступа: [http://en.wikipedia.org/wiki/Pitch\\_\(aviation\)](http://en.wikipedia.org/wiki/Pitch_(aviation)). [Дата обращения 11.05.2014].
9. ROS (операционная система). Свободная энциклопедия Википедия. Режим доступа: [http://en.wikipedia.org/wiki/Robot\\_Operating\\_System](http://en.wikipedia.org/wiki/Robot_Operating_System). [Дата обращения 11.05.2014].

10. Geiger Andreas, Ziegler Julius, Stiller Christoph. StereoScan: Dense 3d Reconstruction in Real-time // Intelligent Vehicles Symposium (IV). 2011.
11. Kitt Bernd, Geiger Andreas, Lategahn Henning. Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme // Intelligent Vehicles Symposium (IV). 2010.
12. Project Tango от Google: смартфон с 3D-сканером пространства. Коллективный блог Habrahabr. Режим доступа: <http://habrahabr.ru/post/213381/>. [Дата обращения 13.05.2014].
13. Project Tango. Project official site. Режим доступа: <https://www.google.com/atap/projecttango/>. [Дата обращения 13.05.2014].
14. Визуальное определение местоположения робота. SMP robotics. Режим доступа: <http://www.smprobotics.ru/technologies/vizualnoe-opredelenie-mestopolozheniya-robota/>. [Дата обращения 13.05.2014].
15. Р.В. Жуков. Методы оптимизации вычисления оптического потока в автономных системах навигации // Наука и образование: научное издание МГТУ им. Н.Э. Баумана Электрон. журн. 2012. № 10. 2014. Режим доступа: <http://sntbul.bmstu.ru/doc/64892.html> [Дата обращения 03.02.2014].
16. Gary Bradski Adrian Kaehler. Learning OpenCV. First Edition изд. O'Reilly, 2008. September.
17. Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы".
18. 2. Санитарно-эпидемиологические правила и нормативы СанПин 2.2.4 1294-03 "Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений".
19. СП 52.13330.2011 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*».



20. Расчет искусственного освещения: Методическое пособие по разделу “Охрана труда” в дипломных проектах. МГТУ им.Баумана. М, 1976.