





# **Реферат**

Данная расчетно-пояснительная записка содержит 131 страницу, 27 иллюстраций, 23 таблицы, 2 приложения, 22 использованных источника.

Ключевые слова: оптический поток, визуальная одометрия, OpenCV, алгоритм Лукаса-Канаде, акселерометр, гироскоп, камера, инерционное измерительное устройство.

Дипломный проект на тему «Подсистема автономного определения перемещения объекта» посвящен разработке автоматизированной системы, которая позволит автономно определять перемещения объекта с использованием видеоряда и данных с инерционных измерительных устройств.

В рамках дипломного проекта были решены следующие задачи:

- исследование предметной области;
- сравнительный анализ аналогов и прототипов;
- организационно-экономическое обоснование разработки;
- разработка структуры и архитектуры системы;
- разработка структур данных;
- разработка алгоритмов обработки информации;
- сравнение производительности различных методов;
- разработка рекомендаций по охране труда.

Графическая часть дипломного проекта содержит 10 листов формата А1.

# Содержание

<b>Нормативные ссылки</b>	<b>7</b>
<b>Определения, обозначения и сокращения</b>	<b>8</b>
<b>1 Конструкторская часть</b>	<b>10</b>
1.1 Разработка технического задания . . . . .	10
1.1.1 Постановка задачи проектирования . . . . .	10
1.1.2 Описание предметной области . . . . .	10
1.1.3 Выбор критериев качества . . . . .	21
1.1.4 Анализ аналогов и прототипов . . . . .	22
1.1.5 Перечень задач, подлежащих решению в процессе разработки . . . . .	31
1.2 Проектирование подсистемы . . . . .	32
1.2.1 Разработка структуры подсистемы . . . . .	32
1.2.2 Разработка формата и структуры данных . . . . .	51
1.2.3 Разработка алгоритмов обработки информации . . . . .	53
<b>2 Технологическая часть</b>	<b>62</b>
2.1 Руководство по интеграции и настройке подсистемы . . . . .	62
2.2 Диаграмма взаимодействия компонентов . . . . .	66
2.3 Диаграмма классов подсистемы . . . . .	66
2.4 Выбор языка программирования . . . . .	66
<b>3 Исследовательская часть</b>	<b>70</b>
3.1 Обоснование области исследования . . . . .	70
3.2 Описание исследуемых методов . . . . .	70
3.2.1 метод Лукаса-Канаде . . . . .	70
3.2.2 метод Horn-Schunck . . . . .	71
3.2.3 метод Farneback . . . . .	71
3.2.4 метод SimpleFlow . . . . .	72
3.2.5 метод Dual TV L1 . . . . .	72
3.3 Проведение эксперимента . . . . .	72
3.3.1 Методика проведения эксперимента . . . . .	72
3.3.2 Исходный код программы . . . . .	73

3.3.3 Тестовый стенд . . . . .	74
3.3.4 Результаты эксперимента . . . . .	74
3.4 Выводы к исследовательской части . . . . .	78
<b>4 Экономическая часть</b>	<b>80</b>
4.1 Обоснование сметы затрат на разработку программного про- дукта ПАОПО . . . . .	80
4.1.1 Расчет затрат на расходные материалы . . . . .	80
4.1.2 Расчет затрат на оборудование . . . . .	81
4.2 Определение трудоемкости выполнения проекта . . . . .	82
4.3 Расчет затрат на оплату труда . . . . .	83
4.3.1 Расчет затрат на страховые взносы . . . . .	85
4.3.2 Расчет затрат на услуги сторонних организаций . . . . .	87
4.3.3 Расчет затрат на накладные расходы . . . . .	87
4.3.4 Расчет прочих расходов . . . . .	87
4.3.5 Итог затрат для заказчика . . . . .	88
4.4 Основные сметы затрат на тестирование, внедрение и эксплу- атацию системы . . . . .	88
4.4.1 Тестирование . . . . .	88
4.4.2 Внедрение и эксплуатация . . . . .	89
4.5 Итого. Расходы на разработку и тестирование . . . . .	90
<b>5 Промышленная экология и безопасность</b>	<b>91</b>
5.1 Анализ опасных и вредных факторов, возникающих при ра- боте на ПЭВМ . . . . .	91
5.1.1 Помещение для работы с ПЭВМ . . . . .	91
5.1.2 Рабочее место оператора и положение за рабочим местом	92
5.1.3 Освещение . . . . .	100
5.1.4 Расчёт освещения . . . . .	102
5.2 Утилизация и списание аппаратных комплектующих . . . . .	107
5.2.1 Стандартная процедура списания и утилизации техники	108
<b>Заключение</b>	<b>110</b>
<b>Список используемых источников</b>	<b>111</b>
Приложение А Техническое задание . . . . .	114

Приложение В Графическая часть . . . . . 121

# **Нормативные ссылки**

В дипломном проекте использованы следующие стандарты:

ГОСТ 19.001-77. «Единая система программной документации. Общие положения»;

ГОСТ 19.105-78. «Единая система программной документации. Общие требования к программным документам»;

ГОСТ 19.106-78. «Единая система программной документации. Требования к программным документам, выполненным печатным способом»;

ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

ГОСТ 9327. «Бумага и изделия из бумаги. Потребительские форматы»;

ГОСТ 1.5. «Государственная система стандартизации РФ. Общие требования к построению, изложению, оформлению и содержанию стандартов»;

ГОСТ 2.105. «Единая система конструкторской документации. Общие требования к текстовым документам»;

ГОСТ 6.38. «Унифицированные системы документации. Система организационно-распорядительной документации. Требования к оформлению документов»;

ГОСТ 7.1. «Система стандартов по информации, библиотечному и издательскому делу. Сведения об источниках»;

ГОСТ 7.12. «Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования»;

ГОСТ 7.32-2001. «Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления»;

СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

# **Определения, обозначения и сокращения**

В расчетно-пояснительной записке использованы следующие Определения, обозначения и сокращения:

API - интерфейс программирования приложений. набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах.

Android - операционная система для смартфонов, планшетных компьютеров, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google и других устройств. Основана на ядре Linux и собственной реализации Java от Google.

C++ - компилируемый статически типизированный язык программирования общего назначения.

GPS - Global Positioning System. Спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат.

Java - объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems.

MATLAB - пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете.

SDK - комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определенного пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

SLAM - Simultaneous Location and Mapping. Метод, используемый роботами и автономными транспортными средствами для построения карты в неизвестном пространстве или для обновления карты в заранее известном пространстве с одновременным контролем текущего местоположения и пройденного пути.

ИИУ - инерционное измерительное устройство. Устройство включающее в себя датчики линейного ускорения (акселерометры) и угловой ск

ности (гироскопы или пары акселерометров, измеряющих центробежное ускорение).

СЛАУ - система линейных алгебраических уравнений. Дрейф нуля - самопроизвольное изменение выходного сигнала со временем при неизменном или даже отсутствующем входном сигнале.

ГЛОНАСС - глобальная навигационная спутниковая система. советская и российская спутниковая система навигации, разработана по заказу Министерства обороны СССР.

Одомтерия - использование данных о движении приводов, для оценки перемещения. Так же термин применяется для любого метода оценки перемещения.

ПЗС - прибор с зарядовой связью, специализированная аналоговая интегральная микросхема, состоящая из светочувствительных фотодиодов, выполненная на основе кремния.

Фреймворк - структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

# **1 Конструкторская часть**

## **1.1 Разработка технического задания**

### **1.1.1 Постановка задачи проектирования**

Целью разработки подсистемы автономного определения перемещения объекта является создание удобного, дешевого и не требующего специализированных устройств для своей работы компонента для интеграции в подвижные автономные системы или в системы управления ими.

### **1.1.2 Описание предметной области**

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

Одной из основных задач в ходе создания подвижных автономных систем является задача определения текущего местоположения объекта в пространстве.

Для этого можно использовать методы имеющие разные области применения и основанные, во-первых, на глобальном позиционировании по спутникам (GPS, ГЛОНАСС), во-вторых, на определении перемещения от стартовой позиции. Так, например, при позиционировании подвижных автономных систем внутри помещения использование спутниковых систем позиционирования становится невозможным, как по причине слабого сигнала или его полного отсутствия, так и из-за недостаточной точности в рамках навигации внутри интерьера помещения. Не стоит забывать и про актуальность систем навигации в космической отрасли, где использование спутников является невозможным в принципе. Методы определения перемещения подвижных автономных систем от стартовой позиции принято называть методами одометрии, которые основываются 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 создали условия к более широкому применению визуальной одометрии. Данный подход позволяет получать один видеоряд через видеокамеру и на его основе получить разные сведения об окружающей среде. Однако метод не лишен недостатков, для устранения которых применяется гибридизация нескольких методов одометрии в рамках одной системы.

Одним из таких подходов является гибридизация методов визуальной одометрии и инерционных измерительных устройств.

При таком подходе данные с видеокамеры и данные с инерционных измерительных устройств обрабатываются параллельно и независимо. В результате два независимых рассчитанных положения носителя сопоставляются и из них выбираются наиболее правдоподобные.

Таким образом, в процессе функционирования спроектированной подсистемы гибридной одометрии протекает следующий процесс. На вход подсистемы 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 расчесывается смещение объекта относительно предыдущего положения и его поворот. Далее эти данные совмещаются и выбираются наиболее правдоподобные, которые затем прибавляются к положению и углу поворота, рассчитанным на предыдущей итерации.

### 1.1.2.2 Графическое представление процесса

Графическое представление процесса представлено на рисунке 1.

### 1.1.2.3 Подсистема автономного определения перемещения объекта в системах управления

Проектируемая подсистема является автономной и не нуждается в управлении со стороны человека. Данное обстоятельство является несомненным плюсом ПАОПО, тем не менее отсутствие человека в контуре управления является и недостатком. Однако проектируемая подсистема яв-

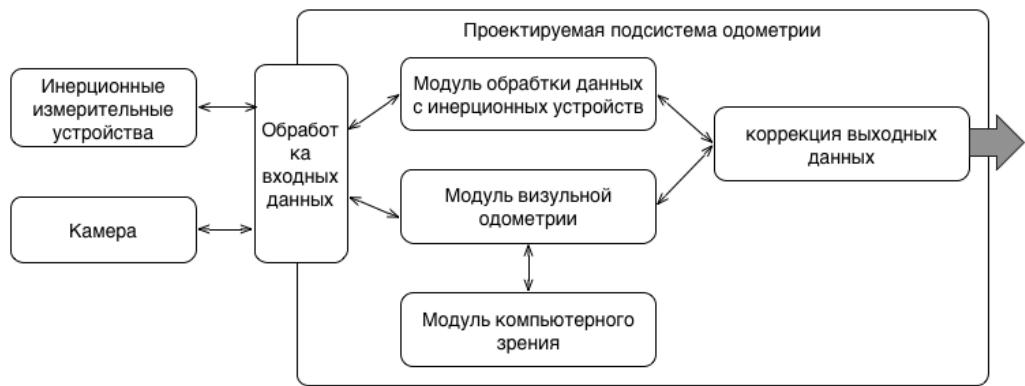


Рисунок 1 – Графическое представление процесса.

ляется важным структурным компонентом в системах управления подвижными объектами, такими как автомобили, роботы, марсоходы и другие. В структуре более высокоуровневых систем, ПАОПО может выполнять следующие функции:

- определение пройденного маршрута;
- помочь оператору при удаленном управлении объектом;
- контроль выполнения управляющих команд;
- автоматическое построение маршрута для возврата в исходную точку.

Кроме того, наличие системы одометрии является обязательным фактором для автоматического построения карты местности и алгоритма SLAM, а так же в целом для систем навигации.

Таким образом можно утверждать, что проектируемая система является важным компонентом для систем управления подвижными объектами. ПАОПО используется подсистемами ручного и автоматического управления в качестве контроля выполнения команд управления в виде обратной связи, а так же в подсистемах навигации для определения текущих координат.

Значимость ПАОПО в системах управления подвижными объектами показан на рисунке 2.

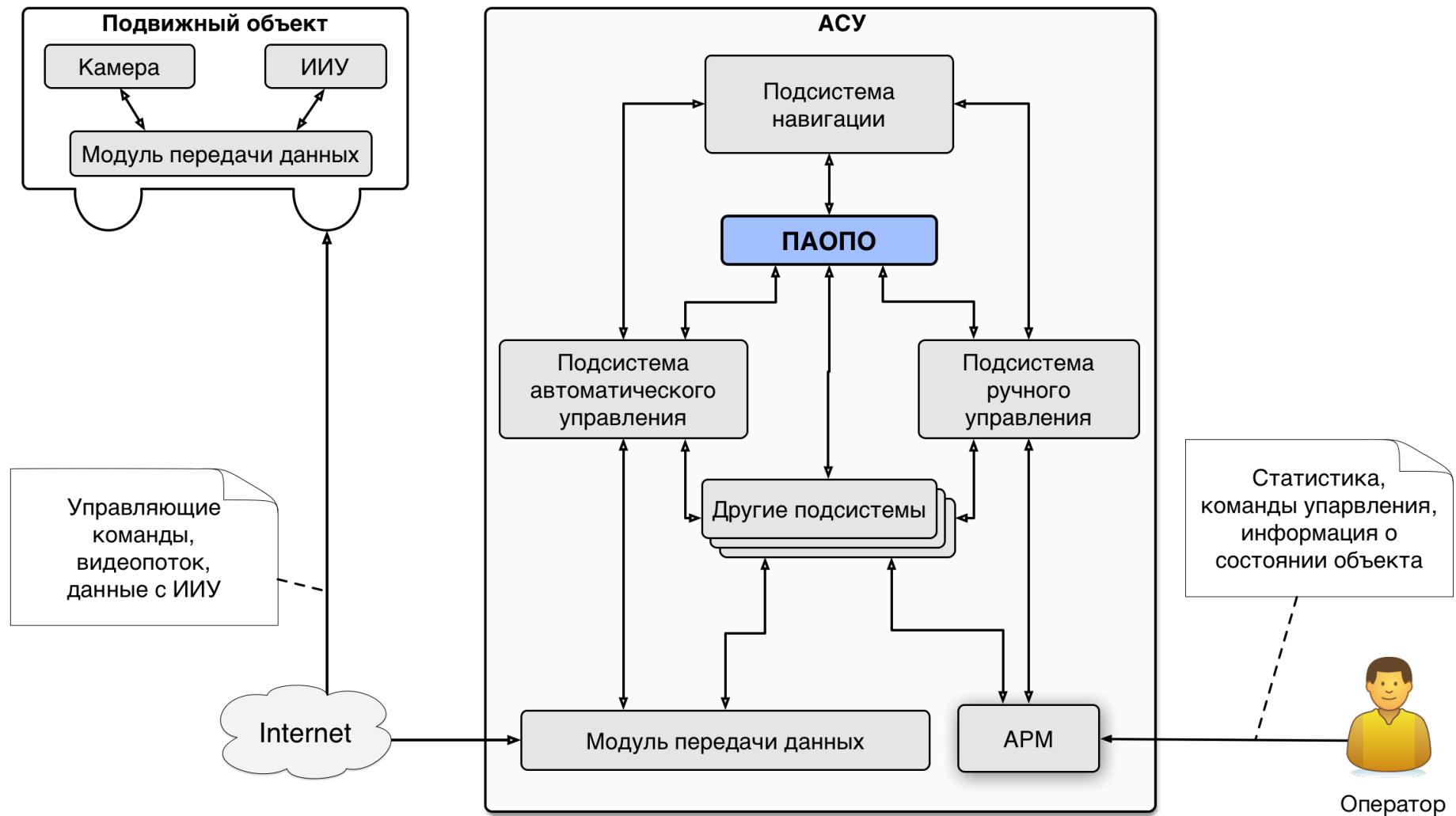


Рисунок 2 – Значимость ПАОПО в системах управления подвижными объектами.

#### 1.1.2.4 Вычисление оптического потока.

Под оптическим потоком будем понимать изображение видимого движения объектов, поверхностей или краев сцены, получаемое в результате перемещения наблюдателя (глаз или камеры) относительно сцены[2]. Пример такого потока показан на рисунке 3.

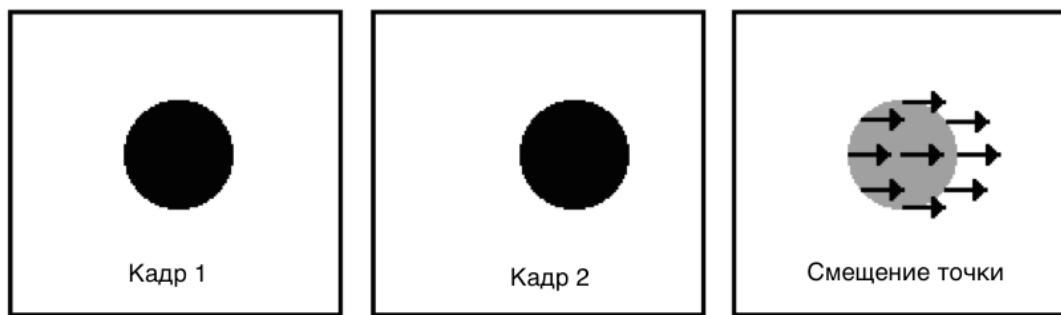


Рисунок 3 – Пример оптического потока.

Существует несколько подходов к определению смещений между двумя соседними кадрами. Например, для каждого небольшого фрагмента (скажем, 8x8 пикселей) одного кадра находится наиболее похожий фрагмент на следующем кадре. В этом случае разность координат исходного и найденного фрагментов позволяет получить требуемое смещение. Основная сложность состоит в том, чтобы быстро отыскать нужный фрагмент, не перебирая весь кадр попиксельно. Различные реализации такого подхода так или иначе решают проблему вычислительной сложности. Платой за скорость является качество. Поэтому в работе рассматривается другой подход, который позволяет получить смещения не для фрагментов, а для каждого отдельного пикселя. Именно с ним в литературе часто связывают термин «оптический поток».

Предлагаемый подход часто называют дифференциальным, поскольку в его основе лежит вычисление частных производных по горизонтальному и вертикальному направлениям изображения. Однако одних только производных недостаточно чтобы определить смещения. Именно поэтому на базе одной простой идеи появилось большое количество методов, каждый из которых использует оригинальный подход, чтобы достичь поставленной цели. В качестве примера приведем метод Лукаса-Канаде (Lucas-Kanade), предложенном в 1981 году Брюсом Лукасом и Такео Канаде. Данный метод

является наименее ресурсоемким[3], обеспечивая при этом 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:

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

При этом  $f_1$  и  $f_2$  можно записать и в общем виде:  $f_1(x) = I(x, y, t)$  ;  $f_2(x) = I(x, y, t + 1)$ .

Связем 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, получим:

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

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:

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

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. 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. Данний недостаток является самым критичным.

### 1.1.2.5 Одометрия с использованием инерциальных измерительных устройств

Навигационные решения надлежащего качества, как показывает практика, могут быть получены 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 геопространственные измерения и вычислительный блок, осуществляющий алгоритмические преобразования данных 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].

Для учета величины «ухода» применяются акселерометры измеряющие линейные ускорения. В равной степени они пригодны и 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 само-

лёта), а также небольшие изменения курса вправо или влево, свойственные судну [6];

- **Крен** — поворот объекта (судна, самолёта, фундамента) вокруг его продольной оси [6];
- **Тангаж** — угловое движение летательного аппарата или судна относительно главной (горизонтальной) поперечной оси инерции [6].

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

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

Обладая чувствительностью к земной гравитации, акселерометры обеспечивают измерение долговременных значений крена и тангажа по схеме, изображенной на рисунке 4. Рассмотрим акселерометр, рабочая ось которого совпадает с продольной осью  $oX$  объекта.

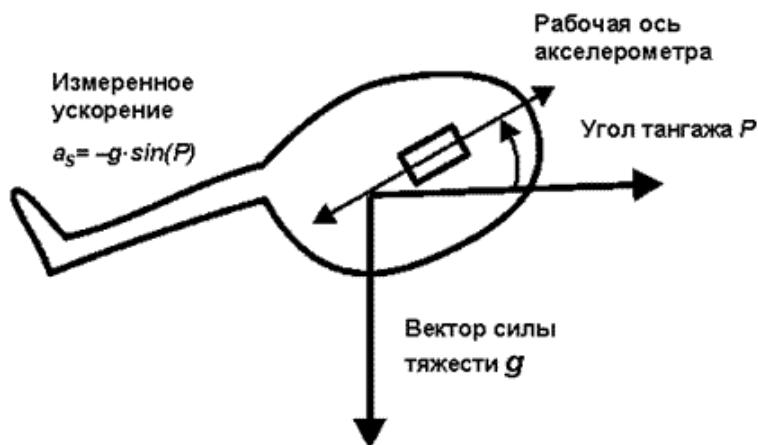


Рисунок 4 – Измерение величины тангажа посредством акселерометров.

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. Полученный результат очевиден, т.к. углы крена и тангажа изначально определены по отношению к вертикали, которая соответствует вектору тяжести (см. рис. 4). Однако описанный метод не может быть использован на практике, так как в приведенной схеме

существенно состояния покоя, в котором должна находиться система. Очевидно, что если это условие не соблюдается, то отсутствует принципиальная возможность выделить вектор ускорения свободного падения из суммы всех ускорений, которую испытывает система.

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

Как отмечено выше, в конструкции навигационного комплекса используются 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 нуля, 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. 5). Точная оценка величины ошибочного углового смещения и его устранение осуществляется при генерации навигационного решения на последующем навигационном уровне.

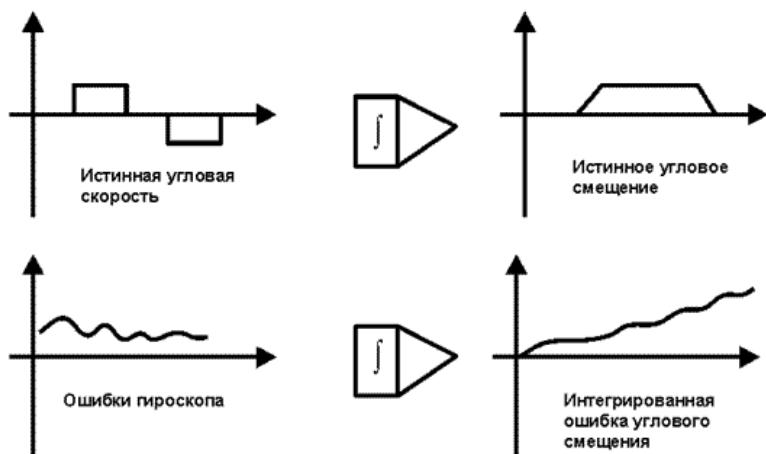


Рисунок 5 – Схема определения углового смещения.

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

Наличие акселерометров позволяет определять величины линейных ускорений, которые испытывает система. Если ориентация системы в физическом пространстве определена точно (с помощью методов описанных выше), то имеется возможность вы-делить 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 правил перехода из физического пространства в географическое, появляется принципиальная возможность оценить гео-дезические координаты системы в любой момент времени.

### 1.1.2.6 Корректировка выходных данных

Так как процессы определения координат с использованием инерциальных устройств и визуальной одометрии проходят параллельно, то на выходе получаются 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 1.1.2.5.

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 видео файла 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:

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

## 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:

$$x / \Delta t \leq V_{\max}^x = V_{\max}$$

$$y / \Delta t \leq V_{\max}^y = V_{\max}$$

$$\alpha / \Delta t \leq V_{\max}^\alpha$$

При выполнении данных неравенств, можно утверждать, что данные правдоподобны. Если же хоть одно из них не выполняется, то к резуль-

татам одометрии следует относиться с определенной долей недоверия и пытаться минимизировать ошибку.

### 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 не 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:

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

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

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

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

Для угла поворота следует учитывать то, что этот показатель может колебаться в пределах от  $0^\circ$  до  $360^\circ$ . Эти неравенства являются еще одними сигнальными параметрами.

#### 1.1.2.7 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 не специальным оборудованием таким, как бытовые камеры.
- **Стоимость необходимого оборудования** – исходя из предыдущего пункта, следует обеспечивать поддержку наиболее дешевого оборудования (менее 100\$).
- **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 подвижных автономных систем 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 свободность 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 распределенной работы. ROS был первоначально разра-

ботан 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, в то время 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.[7]

В состав ROS так же входят различные пакеты одометрии, наиболее приемлемым является пакет LIBVISO2 [8].

LIBVISO2 (библиотека для визуальной одометрии, версия 2) является очень быстрой кроссплатформенной библиотекой, написанной на C++ с MATLAB оберткой для использования. Основная цель библиотеки – вычисление передвижения движущейся моно- или стереокамеры. Не будем подробно рассматривать стереоверсию библиотеки, так как этот класс одометрии лежит в другой плоскости. Монокулярная версия на данный момент находится в статусе экспериментальной. Кроме того, она 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 разрешает 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[9]. Кроме того, существует еще одна проблема: когда камера выполняет только чистое вращение, матрицы для решения СЛАУ вырождаются (даже если система очень мощная) и алгоритм перестает корректно работать.

#### 1.1.4.2 Project Tango

Небольшая группа инженеров ATAP (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 ATAP сконструировала 5-дюймовый смартфон, оснащенный 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.

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 [10][11].

#### 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 сохраненным 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 предет 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. Объем 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 радиотени, 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 [12].

#### 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 количественные. 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

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

Описание	Данную систему сложно интегрировать с другими системами, потребуются 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

Введем весовой коэффициент, отображающий важность параметра. Для выбора значений весовых коэффициентов воспользуемся методом анализа иерархий. Для этого составим матрицу парных сравнений всех критериев и сравним их значимость между собой, используя шкалу Саати. Результат такого сравнения отражен в таблице 7.

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 накапливание 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 proprietарной
Лингв. оценка	Да	Нет
Балл	1	0,4

Таблица 7 – Сравнение критериев методом анализа иерархии.

	K1	K2	K3	K4	K5	$C_i$
K1	1	1	3	3	5	2,141
K2	1	1	3	3	5	2,141
K3	0,33	0,33	1	1	3	0,8
K4	0,33	0,33	1	1	3	0,8
K5	0,2	0,2	0,33	0,33	1	0,337

$$\sum C_i = 6,218$$

$$\alpha_i = \frac{C_i}{\sum C_i}$$

Таким образом весовые коэффициенты получились равными:

$$\alpha_1 = 0,344$$

$$\alpha_2 = 0,344$$

$$\alpha_3 = 0,129$$

$$\alpha_4 = 0,129$$

$$\alpha_5 = 0,054$$

Оценим согласованность матрицы сравнения критериев:

$$OC = \frac{(5,034 - 5)}{(5 - 4) \cdot 1,12} = 0,0075 \leq 0,1$$

Так как оценка согласованности получилась меньше 0,1, то можно утверждать, что матрица является согласованной. и данным расчетам можно верить.

Сумма весовых коэффициентов по всем критериям качества равна 1.

Таким образом, можно составить сравнительную таблицу, содержащую количественные и качественные оценки и подсчитать итог, методом взвешенной суммы. Итоговое сравнение с учетом весового коэффициента и количественных оценок приведены в таблице 8.

### **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;
2. разработка математического обеспечения для функционирования подсистемы;
3. разработка требований к формату и структуре передаваемых данных;

Таблица 8 – Сравнение аналогов. Количественные оценки, весовой коэффициент.

Критерий	Вес. коэф.	Проектируемая подсистема	Аналоги		
			ROS odometry	Project tango	SMP Robotics
Необходимость в специальном оборудовании	0,344	1	1	0,4	0,4
Стоимость необходимого оборудования	0,344	1	0,6	0,6	0,1
Сложность интеграции	0,129	1	0,1	1	0,6
Точность	0,129	0,6	1	1	1
Возможность свободного использования	0,054	1	1	1	0,4
Итого:		0,9484	0,7463	0,656	0,4

4. разработка алгоритмов обработки данных;
5. выбор и обоснование КТС, необходимого для реализации системы;
6. оценка предполагаемого качества функционирования системы;
7. организационно-экономическое обоснование разработки;
8. 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 6.  
**Назначение модулей**

- **Модуль обработки входных данных** - нормализует входные данные и аккумулирует их, в случае слишком высокой частоты их поступления.



Рисунок 6 – Структура компонентов.

- **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 1.1.2.6).
- **Модуль визуальной одометрии** – вычисляет перемещения на основе видео ряда. 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[13].

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 7.

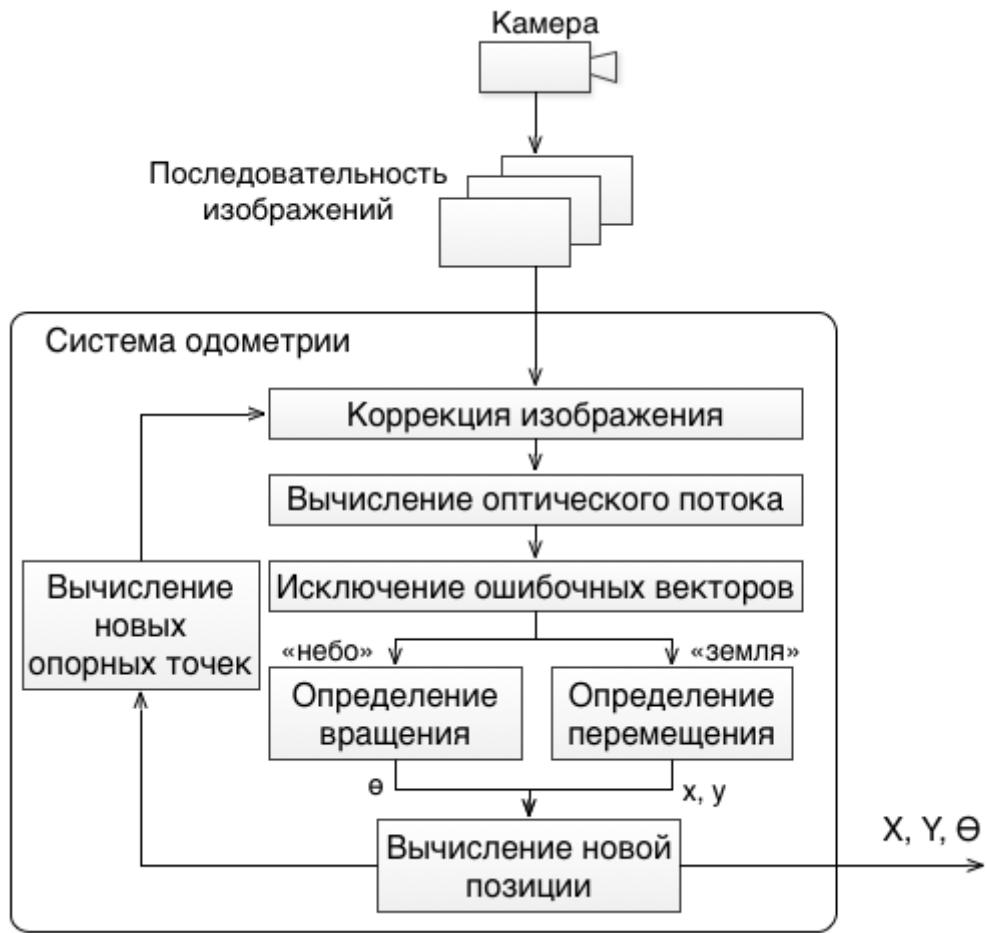


Рисунок 7 – Графическое представление процесса визуальной одометрии

### 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  $Z_a$ , которые показывают ускорения по осям  $X$ ,  $Y$ ,  $Z$ .

Известно, что  $\int_a^b a(t)dt = V(t) + C$ . В то же время  $\int_a^b V(t)dt = S(t) + C$ . Таким образом получаем:

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

Если  $C_s = 0$  - это начальное положение носителя, то:

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

Однако все упрощается, если период замеров настолько мал, что ускорение можно считать равномерным. Тогда формулы приобретают 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. 8): показывающими рассчитанное значения перемещения и скорости в зависимости от времени.

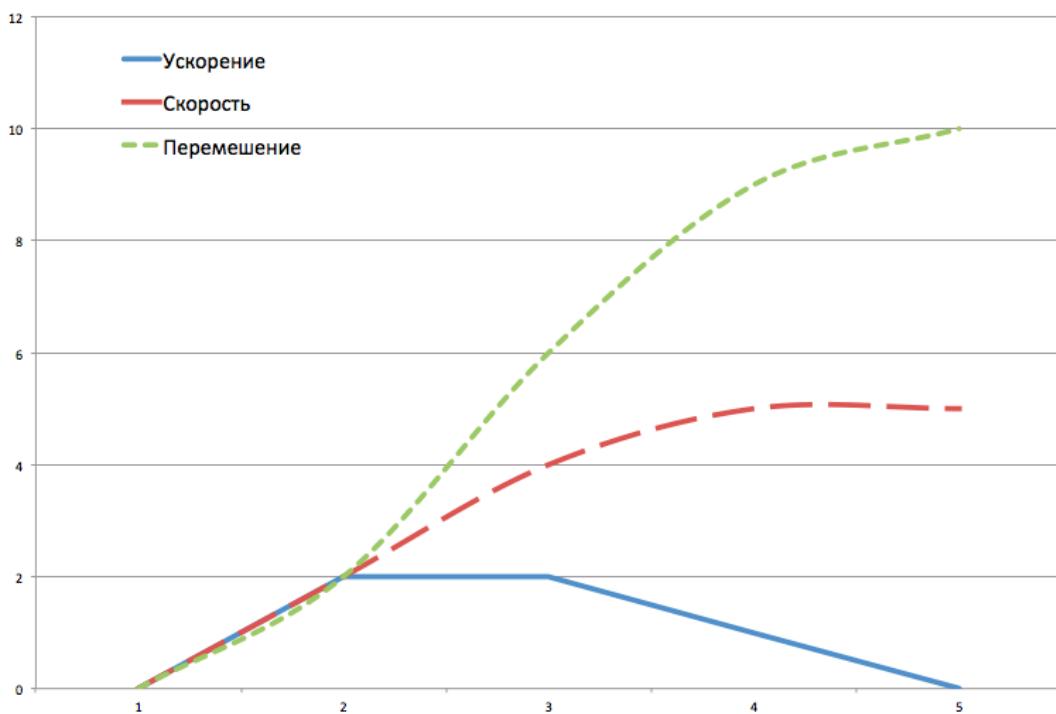


Рисунок 8 – Пример интегрирования ускорения для получения перемещения

Производя подобные математические вычисления для координат  $X$  и  $Y$  получаем перемещения по этим координатам. Причем можно 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 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 выбирать точку на большой однотонной стене, то найти ее на следующем кадре видео будет крайне не легко.

Если все точки на стене будут одинаковыми или даже очень похожи, то вероятность отслеживания этой точки в последующих кадрах будет не высокой. С другой стороны, если мы выберем точку, которая является уникальной, то найти ее на последующих кадрах не составит труда. На практике, точка или черта, которую выбирают в качестве ключевой должна быть уникальной или почти уникальной, и должна быть параметризируемой таким образом, чтобы ее можно было отличить от других точек на изображении.

Возвращаясь к аналогии с большой однотонной стеной, следует попытаться искать точки, которые имеют некоторые значительные изменения в своих окрестностях, например, большое значение производной. Правда на практике оказывается, что этого не достаточно, так как такая точка, 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 с высокой вероятностью эта точка будет уникальной на протяжении нескольких кадров. 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 ( $\partial^2x, \partial^2y, \partial x\partial y$ ) интенсивностей изображения. Такие производные связаны с матрицей 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[14].

Вычисляя оптический поток, мы можем связать с каждым пиксели 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 - выборочному оптическому потоку. Алгоритмы такого рода опираются на некоторые средства определения подмножество точек, которые должны отслеживаться. Если эти точки имеют определенные свойства, такие как свойства ключевых особенностей, обсуждаемые ранее, тогда отслеживание будет относительно точным и надежными. Для многих практических случаев, вычислительная стоимость выборочного оптического потока намного меньше, чем у плот-

ного оптического потока, поэтому последнему в последнее время 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 наивысшей 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 вид, переходя от кадра к кадру. Для изображений в градациях серого это означает, что яркость пикселя не меняется, при его отслеживании от кадра к кадру. Для случая с цветными изображениями есть более строгое допущение, однако оно сводится к случаю изображений в градациях серого.

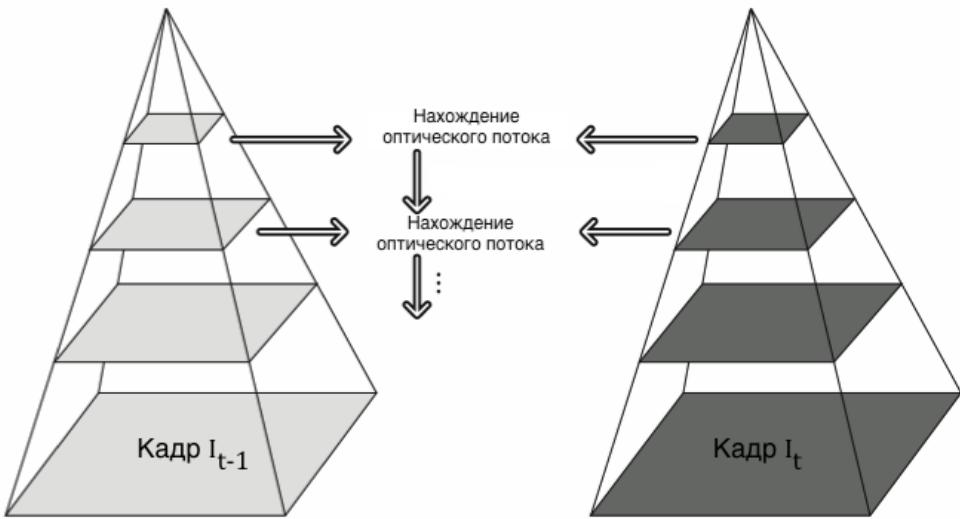


Рисунок 9 – Графическое представление работы "пирамидальной" версии алгоритма ЛК

2. *Малые сдвиги.* Изображение двигается медленно во времени. На практике это означает, что объект от кадра к кадру сдвигается незначительно.
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 диффе-

ренцирования:

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

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 10 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.

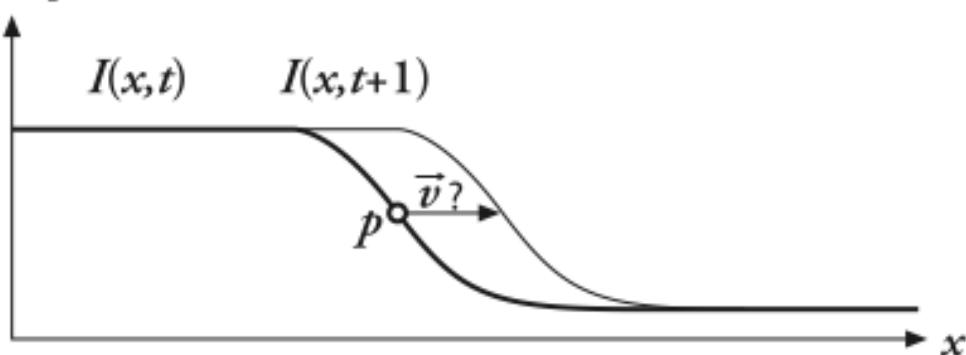


Рисунок 10 – Графическое представление задачи нахождения оптического потока в одномерном случае.

На рисунке 11 показано как можно решить такую задачу. При учете двух первых предположений получим следующее:

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

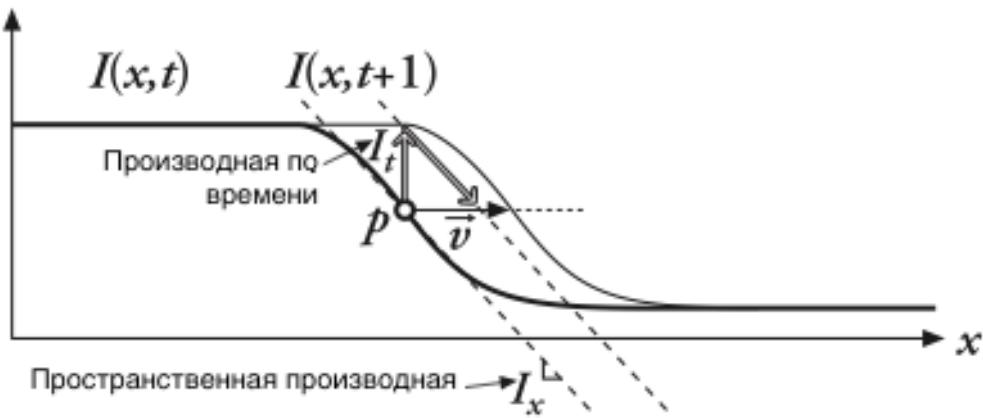


Рисунок 11 – Графическое представление решения задачи нахождения оптического потока в одномерном случае.

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  $5 \times 5$  пикселей, получив 25 уравнений. Очевидно, что 3 допущение не всегда верно, поэтому в общем случае система не имеет решения.

Попробуем решить эту систему к минимизации функции 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. Самые распространенный вариант – двумерная гауссиана, которая дает наибольший вес центральному пикセルю и все меньший по мере удаления от центра (см. рис. 12).

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 компактной форме, приравняем к 0:

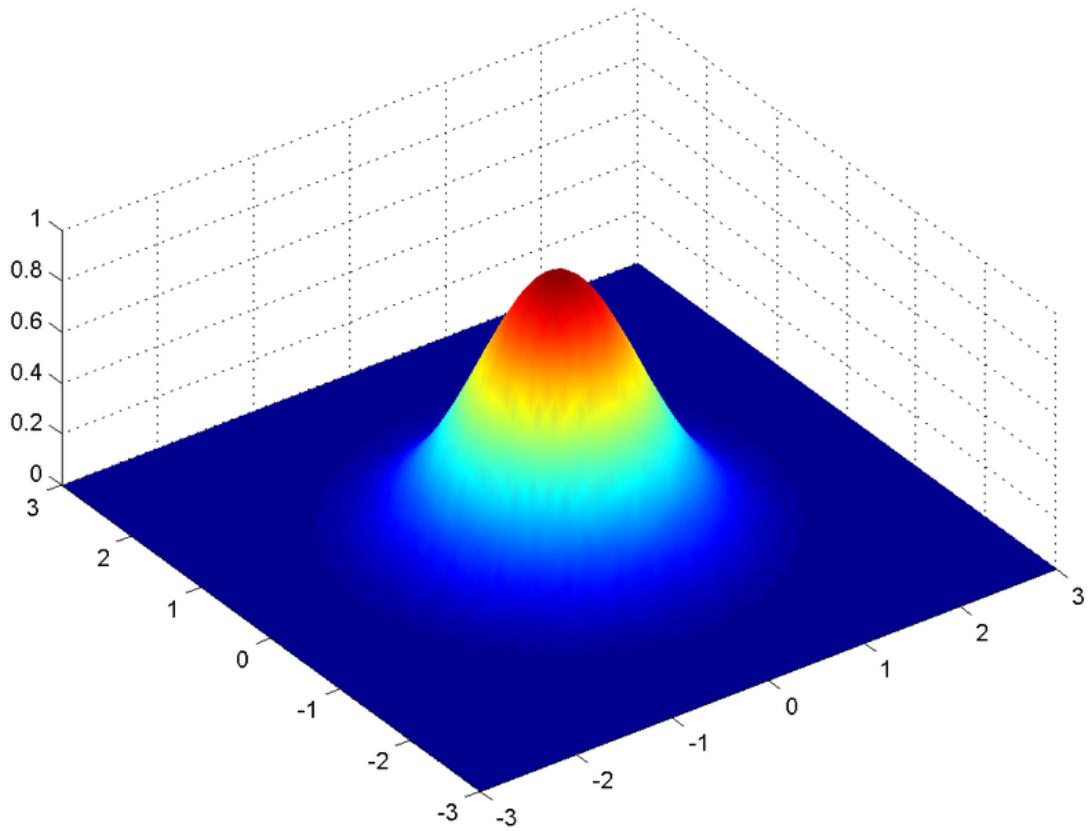


Рисунок 12 – Двумерная гауссиана

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

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

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 отснятым ей по отснятым ей фотографиям или видео[15].

Рассмотрим простейшую модель камеры - камеры-обскуры. В этой простой модели, свет рассматривается в качестве потока от сцены или удаленного объекта, но только один луч попадает из любой точки этого объекта 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 - ее фокусным расстоянием. На рисунке 13 схематично показана рассматриваемая модель камеры[14].

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 с получим:

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

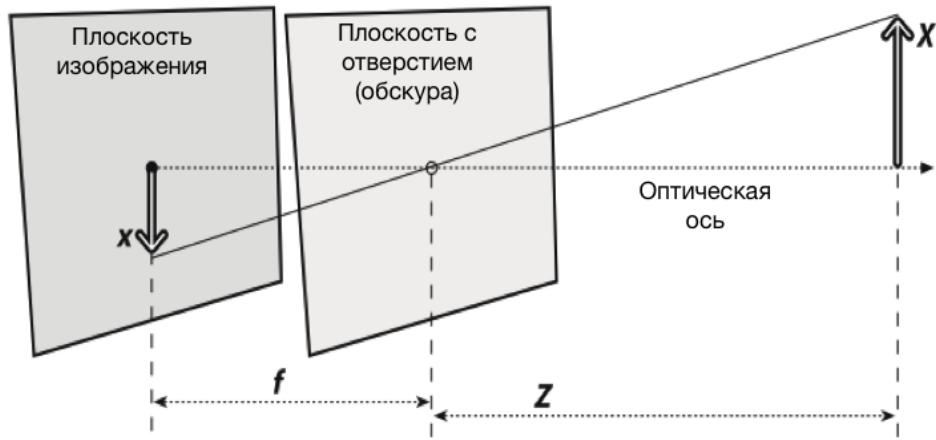


Рисунок 13 – Модель камеры-обскуры

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. Запишем эти уравнения в матричной форме:

$$q = \frac{1}{Z} \cdot M \cdot Q \quad (2)$$

где  $Q_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$  - координаты точки во внешней системе координат,

$q_i = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$  - координаты проекции точки,

$$M = \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix} \text{ - матрица проекции.}$$

Теперь введем линзу в модель камеры. Линза так же вносит искажения, природа происхождения которых показана на рисунке 14.

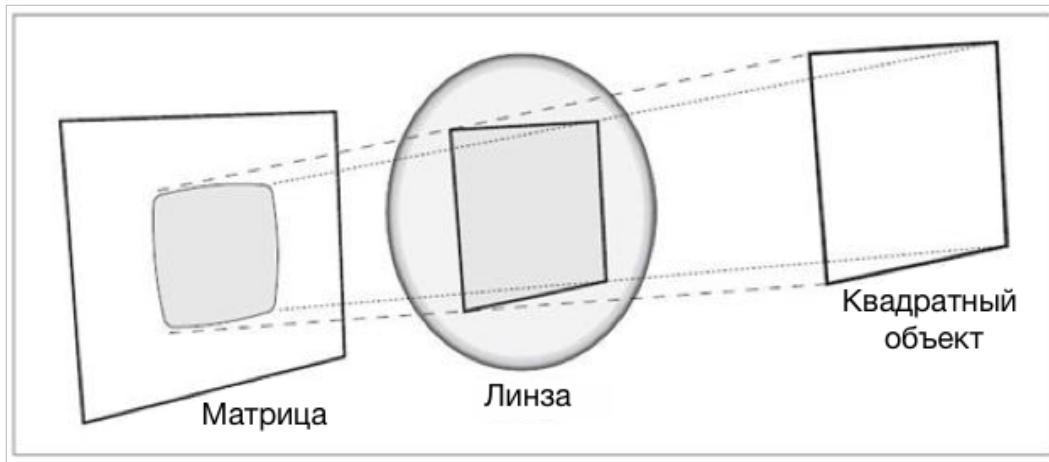


Рисунок 14 – Искажения изображения, вносимые линзой

Данный вид искажения называется радиальным. Он учитывается следующим образом:

$$u_{corrected} = u \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4)$$

$$v_{corrected} = v \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4)$$

Здесь  $r$  – расстояние от точки пересечения главной оптической осью матрицы до точки проекции;  $k_1, k_2$  – коэффициенты радиального искажения.

С учетом этих искажений формула 2 принимает вид:

$$q = \frac{1}{Z} \cdot \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot M \cdot Q \quad (3)$$

где  $\lambda$  - корректирующая функция:

$$\lambda = 1 + k_1 \cdot r^2 + k_2 \cdot r^4$$

Кроме радиального искажения существует ещё тангенциальное. Оно возникает из-за того, что плоскость матрицы не перпендикулярна главной оптической оси. Но оно очень мало, и его можно не учитывать.

При установке камеры тоже появляются погрешности, которые необходимо учесть в модели.

Погрешности ориентации:

$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{pmatrix}$$

$$R_y(\varphi) = \begin{pmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{pmatrix}$$

Результирующая матрица вращения камеры вокруг требуемой системы координат:

$$R(\psi, \varphi, \theta) = R_x(\psi) \cdot R_y(\varphi) \cdot R_z(\theta)$$

, где  $\theta, \varphi, \psi$  - углы Эйлера.

Вектор смещения камеры относительно требуемой системы координат:

$$T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

Теперь модель камеры с учётом радиального искажения и погрешностей установки выглядит так[16]:

$$q = \frac{1}{Z} \cdot \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot M \cdot R \cdot (Q + T)$$

### Вычисление перемещения и угла поворота объекта на основе видеопотока

Для перехода от векторов оптического потока к векторам перемещения объекта обязательна предварительная калибровка камеры и ее жесткое закрепление на определенной высоте на носителе. При этом камера закрепляется таким образом, чтобы в кадре 70% занимало изображение поверхности, по которой происходит перемещение. Иными словами, линия горизонта должна находиться на уровне 3/10 высоты кадра от ее верхнего края. Готовую к визуальной одометрии подвижную платформу схематично можно изобразить так, как показано на рисунке 15.

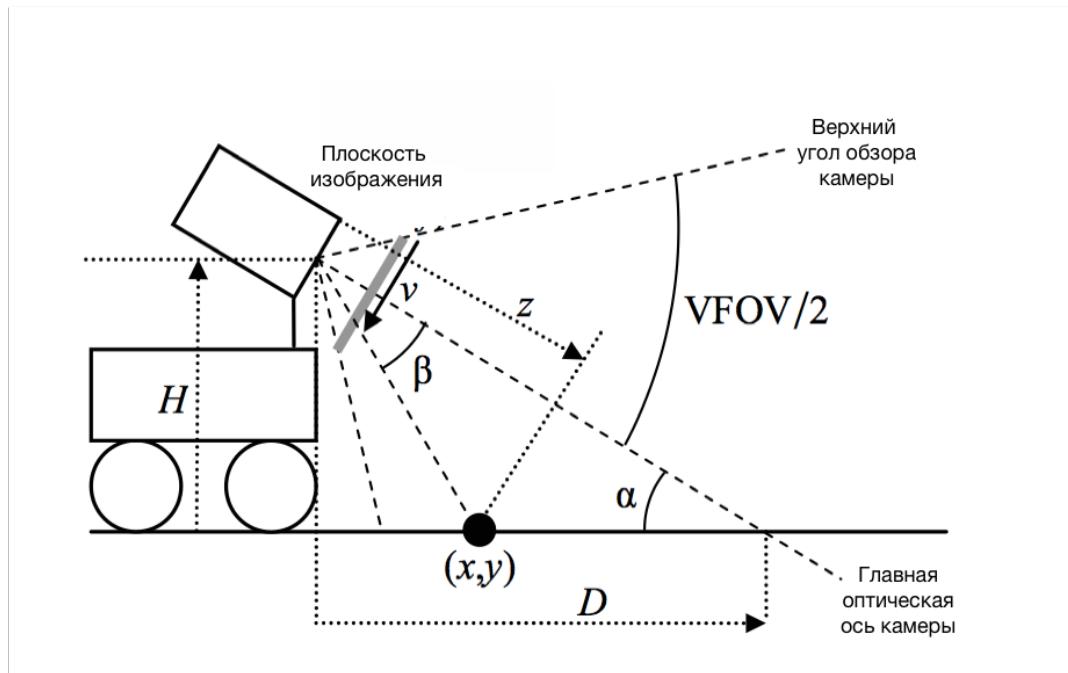


Рисунок 15 – Схема расположения камеры на подвижном объекте.

После коррекции изображение будет восприниматься так, как будто оно расположено в плоскости  $v$ , перпендикулярной основному вектору на-

правления камеры. Далее это изображение делится на 2 части: часть, отображающую объекты над горизонтом, и часть, отображающую объекты на плоскости движения.

Теперь, зная  $H$  - высоту закрепления камеры и  $D$  - расстояние до пересечения ее главной оптической оси с поверхностью передвижения, можно высчитать положение точек изображения на поверхности следующим образом:

$$\tan(\alpha) = H/D,$$

$$\tan(\beta) = (2v - V) \cdot (VFOF/2),$$

где  $V$  - высота изображения,  $v$  - расстояние от верхнего края изображения до исследуемой точки изображения,  $VFOF$  - вертикальный угол обзора камеры. Тогда:

$$y = \frac{H}{\tan(\alpha + \beta)}, \quad (4)$$

$$z = \frac{H \cos(\beta)}{\sin(\alpha + \beta)},$$

где  $y$  - расстояние от робота до точки на плоскости, а  $z$  - глубина точки (см. рисунок 15).

Для определения вращения камеры необходимо вычислить расстояние до линии горизонта кадра по формуле 4. Если считать, что все точки выше этой линии находятся на таком расстоянии  $y_{\text{гор}}$ , то переходя к цилиндрической системе координат с помощью формулы:

$$\varphi = \arctan\left(\frac{y}{x}\right)$$

получается разницу между ключевыми точками двух кадров в цилиндрической системе координат в градусах. Эта разница и будет равна углу поворота камеры.

### 1.2.2 Разработка формата и структуры данных

В ходе функционирования подсистемы информация поступает на ее вход, передается между модулями внутри системы, после чего поступает на выход. При этом эта информация носит разные характер и смысл. Так

как подсистема носит программный характер, то все потоки данных реализуются исключительно программными средствами, в виде структур данных. При этом сетевые технологии не накладывают никаких ограничений на форматы информационных структур.

Дадим описание основных форматов обмена информацией.

- **Видео (вход).** Видео подается на вход системы покадрово, в связи с чем алгоритм кодирования и сжатия видео не существенен, так как задача формирования следующего кадра в его полном объеме ложиться на вызывающую сторону. По сути, каждый кадр представляет собой 1 картинку в формате JPEG. Такую картинку проще всего представать 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, трех осевого акселерометра. Данные имеют цифровое представления, дробное.
- **Угловое ускорение по трем осям (вход).** Данные подаваемые с гироскопов или одного, трех осевого гироскопа. Данные имеют цифровое представления, дробное.
- **Текущие координаты и угол поворота (выход).** Вычитанное 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 элемен-

тами 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**

Высокоуровнево, алгоритм работы подсистемы выглядит следующим образом:

1. установить текущее положение равным  $(0; 0)$ , угол, указывающий направление движения, равным  $0^\circ$ ;
2. получить изображение и данные с инерционных измерительных приборов;
3. изменить размер изображения, привести его к нужному формату, устранить искажения в изображении;
4. вычислить перемещения и вращение объекта на основе данных с инерционных измерительных устройств;
5. вычислить перемещения и вращения камеры на основе визуальной одометрии;
6. сравнить данные, полученные на этапах 4 и 5, отбросить менее правдоподобные.

7. прибавить перемещения по осям  $oX$  и  $oY$ , а так же поворот относительно вертикальной оси к предыдущим значениям положения и угла направления движения;
8. вернуть данные о текущем положении и ориентации;
9. повторить пункты 2-8 для всех поступающих данных.

Графическое представление общего алгоритма работы в виде блок-схемы представлено на рисунке 16.



Рисунок 16 – Блок-схема общего алгоритма работы подсистемы.

### 1.2.3.2 Алгоритм визуальной одометрии

В общем случае алгоритм визуальной одометрии можно описать следующим образом:

1. выявить ключевые точки изображения;
2. если нет предыдущего изображения, то сохранить текущее для последующих итераций и перейти к шагу 10;
3. вычислить оптический поток;
4. исключение ошибочных векторов;
5. разделение оптического потока на две части: «наземную» и «небесную»;
6. в «небесной» части переход к цилиндрической системе координат и вычисление угла поворота  $\theta$  относительно двух последних кадров;
7. в «наземной» части выделить векторы из оптического потока и вычислить перемещение в плоскости  $x - y$ , получив вектор  $(x, y)$ ;
8. прибавить полученный вектор и угол поворота к изначальному положению объекта, получив новое положение объекта  $(X, Y, \Theta)$ ;
9. вернуть новое положение объекта;
10. сохранить текущее изображение для следующей итерации;
11. перейти к шагу 1 для следующего кадра;

Однако пункт 8 справедлив для систем визуальной одометрии в целом, в текущей же работе эта обязанность перекладываются к модулю корректировки выходных данных всей системы в целом. Алгоритм визуальной одометрии, использующийся в данной работе представлен на рисунке 17.

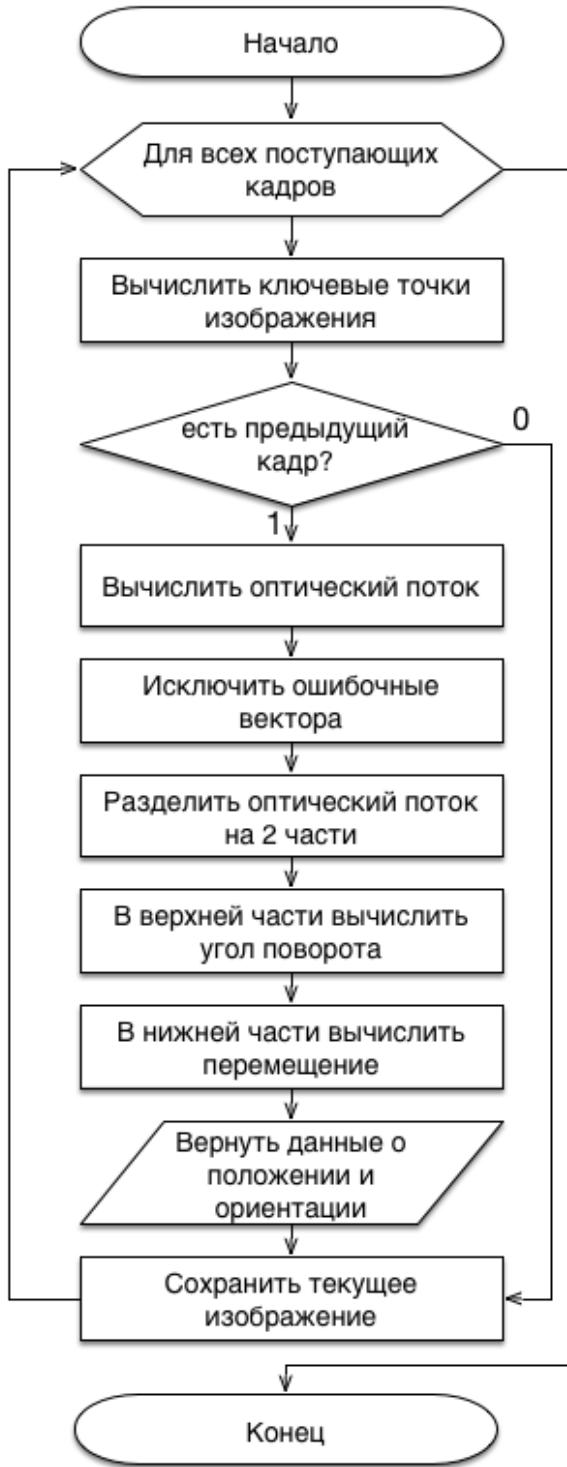


Рисунок 17 – Блок-схема алгоритма визуальной одометрии.

### 1.2.3.3 Алгоритм обработки данных с ИИУ

Алгоритм обработки данных с инерционных измерительных устройств довольно примитивен и состоит из следующих шагов:

1. инициализация начальных значений:

- начальная скорость  $V_{\text{нач}} = V_{\text{пред}} = 0$ ;
- начальное ускорение  $a_{\text{нач}} = 0$ ;
- текущее время  $t_0 = t_{\text{пред}} = 0$ ;
- начальный угол направления  $\omega_{\text{пред}} = 0$ .

- измерение прошедшего времени  $\Delta t = t_{\text{тек}} - t_{\text{пред}}$ ;
- интегрирование данных с акселерометров для получение изменения скоростей и вычисление текущей скорости:

$$V_{\text{тек}} = \int_t a + V_{\text{пред}} = \frac{a_{\text{пред}} + a_{\text{тек}}}{2} \cdot \Delta t + V_{\text{пред}};$$

- интегрирование полученных скоростей для вычисления перемещения:

$$S = \frac{V_{\text{пред}} + V_{\text{тек}}}{2} \cdot \Delta t;$$

- интегрирование данных с гироскопов для получения угла поворота:
- обновление полученных предыдущих значений текущими;
- для всех поступающих данных повторять пункты 2-6.

Блок-схема данного алгоритма представлена на рисунке 18.



Рисунок 18 – Блок-схема алгоритма одометрии на основе инерционных измерительных приборов.

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

В основе модуля корректировки выходных данных лежит простой алгоритм проверки данных, полученных с помощью визуальной одометрии, путем вычисления сигнальных параметров описанных в пункте 1.1.2.6.

При «срабатывании» хотя бы одного из них, все вышедшие из нормы показатели рассчитываются для данных полученных на основе инерционных измерительных устройств. Далее, из двух вариантов выбирается один, наиболее правдоподобный и менее ошибочный.

Если же не один из сигнальных параметром не сработал, на выход подаются данные, полученные с помощью визуальной одометрии.

Графическое представление данного алгоритма в виде блок-схемы представлено на рисунке 19.

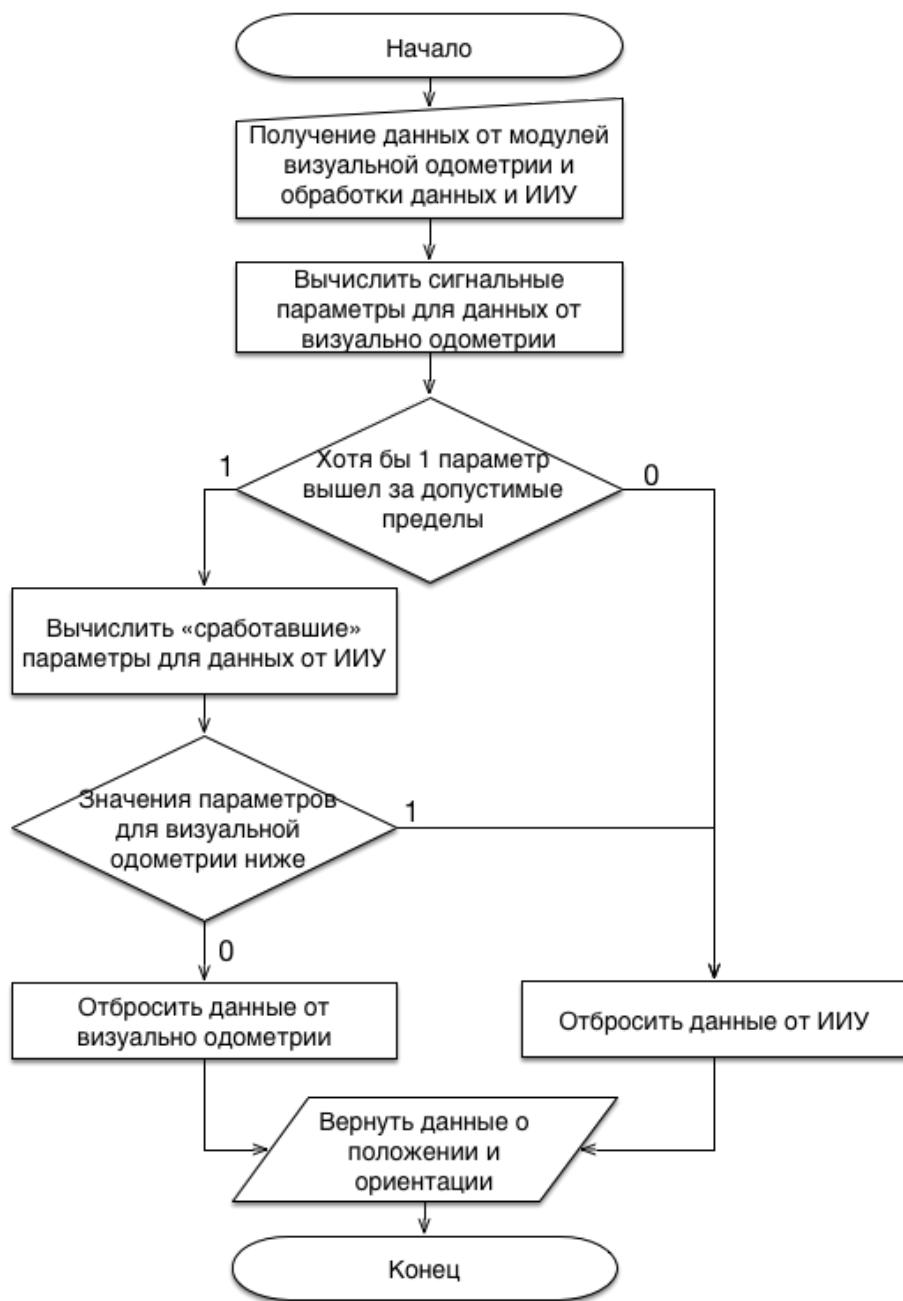


Рисунок 19 – Блок-схема алгоритма модуля корректировки выходных данных.

## **2 Технологическая часть**

### **2.1 Руководство по интеграции и настройке подсистемы**

Одной из целей при создании ПОАПО являлось создание удобного и простого в интеграции с другими системами модуля для автономной навигации объектов. Руководствуясь этой целью, процесс установки и настройки был максимально упрощен. Для корректной работы подсистемы необходимо использовать оборудование, указанное в техническом задании к системе, а именно:

- требования к цифровой камере:
  - разрешение не ниже 1 Мп;
  - интерфейс соединения со скоростью не ниже 12 Мбит/с;
  - жесткое крепление на объекте;
- Требования к инерциальному измерительному устройству
  - трехосевой гироскоп с диапазоном измерения до  $2000\text{ }^{\circ}/\text{s}$  и точностью не ниже, чем  $0,2^{\circ}$  на  $1\text{ }^{\circ}/\text{s}$ ;
  - акселерометр с тремя степенями свободы и диапазоном измерения  $\pm 10g$ .

Вторым этапом интеграции является жесткое закрепление камеры и ИИУ на подвижном объекте, перемещения которого и предполагается рассчитывать. Камеру необходимо расположить по направлению движения и так, чтобы в кадре 70% изображения занимала горизонтальная поверхность, по которой происходит движение, и 30% кадра занимали объекты, расположенные над линией горизонта.

Далее нужно провести калибровку камеры для выявления внутренних и внешних параметров камеры. С теоретической точки зрения данных

процесс описан в пункте 1.2.1.4. На практике процесс является более простым и заключается в фиксации «специального изображения» камерой и последующем анализе полученного изображения. Данный метод калибровки называется «новым гибким методом калибровки камеры» и был предложен Zhengyou Zhang[17]. Как правило в качестве «специального изображения» используется изображения шахматной доски. Суть метода заключается в нахождении углов такой доски и подбору внутренних параметров камеры таким образом, чтобы на изображении эти углы выстраивались в прямые линии. Для корректного определения параметров камеры желательно получить как можно больше изображений такой доски под разными углами. Схематично данный процесс отображен на рисунке 20.

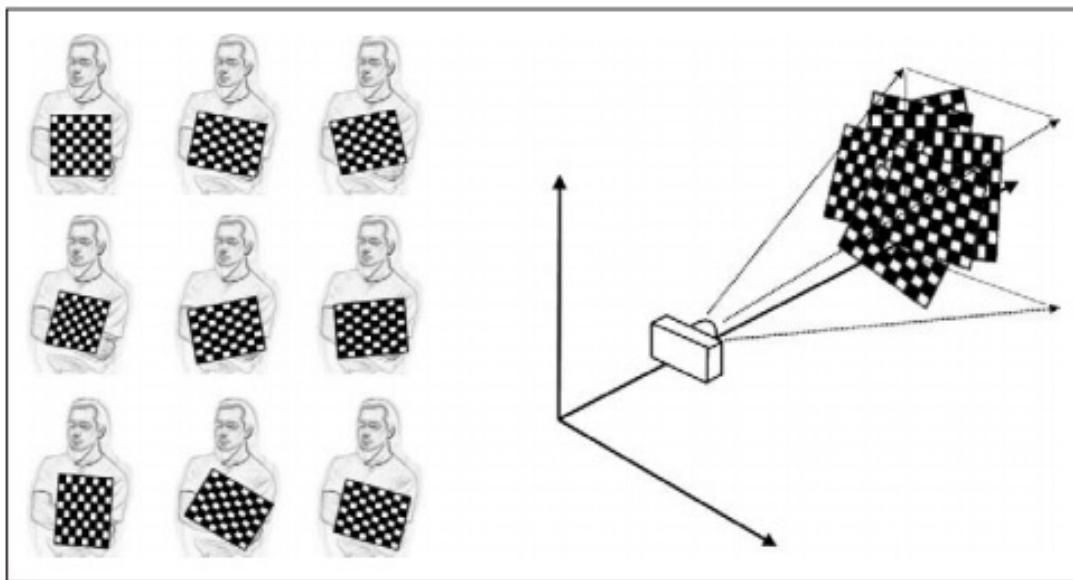


Рисунок 20 – Процесс гибкого метода калбровка камеры.

В то же время, данный метод можно применять для вычисления матрицы трансформации изображений. Рассмотрим рисунок 21. Если фиксировать изображения шахматной доски в плоскости левой фигуры, то при удалении искажений мы получим изображение строго перпендикулярное камере. В визуальной одометрии наоборот необходимо отобразить все ключевые точки в плоскости пола. Таким образом, получив матрицу изменения изображения, транспонируем ее и будем применять для нормализованных изображений с целью определить реальное положение ключевых точек кадра на плоскости перемещения.

После установки необходимого оборудования и его калибровки необходимо согласовать программный интерфейс подсистемы для корректно-

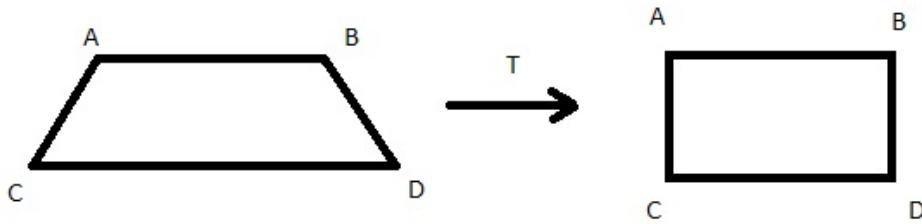


Рисунок 21 – Принцип перспективного преобразования изображений.

го передачи видеоряда и данных с ИИУ. Подсистема ПАОПО предоставляет следующий API:

```
PutNewData(Frame currentFrame, Aceleration[] acel,
Rotation[] rotation, callback onResultAvailable
[, timestamp currrentTime])
```

- ***Frame currentFrame*** - очередной кадр;
- ***Aceleration[] acel*** - массив ускорений по осям  $oX$ ,  $oY$ ,  $oZ$ ;
- ***Rotation[] rotation*** - массив угловых скоростей вокруг осей  $oX$ ,  $oY$ ,  $oZ$ ;
- ***callback onResultAvailable*** - функция обработчик результата;
- ***timestamp currrentTime*** - уникальная метка набора данных. Используется для идентификации выходных данных.

Пользователю необходимо реализовать свой обработчик *onResultAvailable* и передать ссылку на этот метод в функцию *PutNewData*. По сигнатуре метод должен быть следующим:

```
onResultAvailable(int dx, int dy, int dAngle)
```

- ***int dx*** - перемещение по оси  $oX$ ;
- ***int dy*** - перемещение по оси  $oY$ ;
- ***int dAngle*** - угол поворота вокруг оси  $oZ$ .

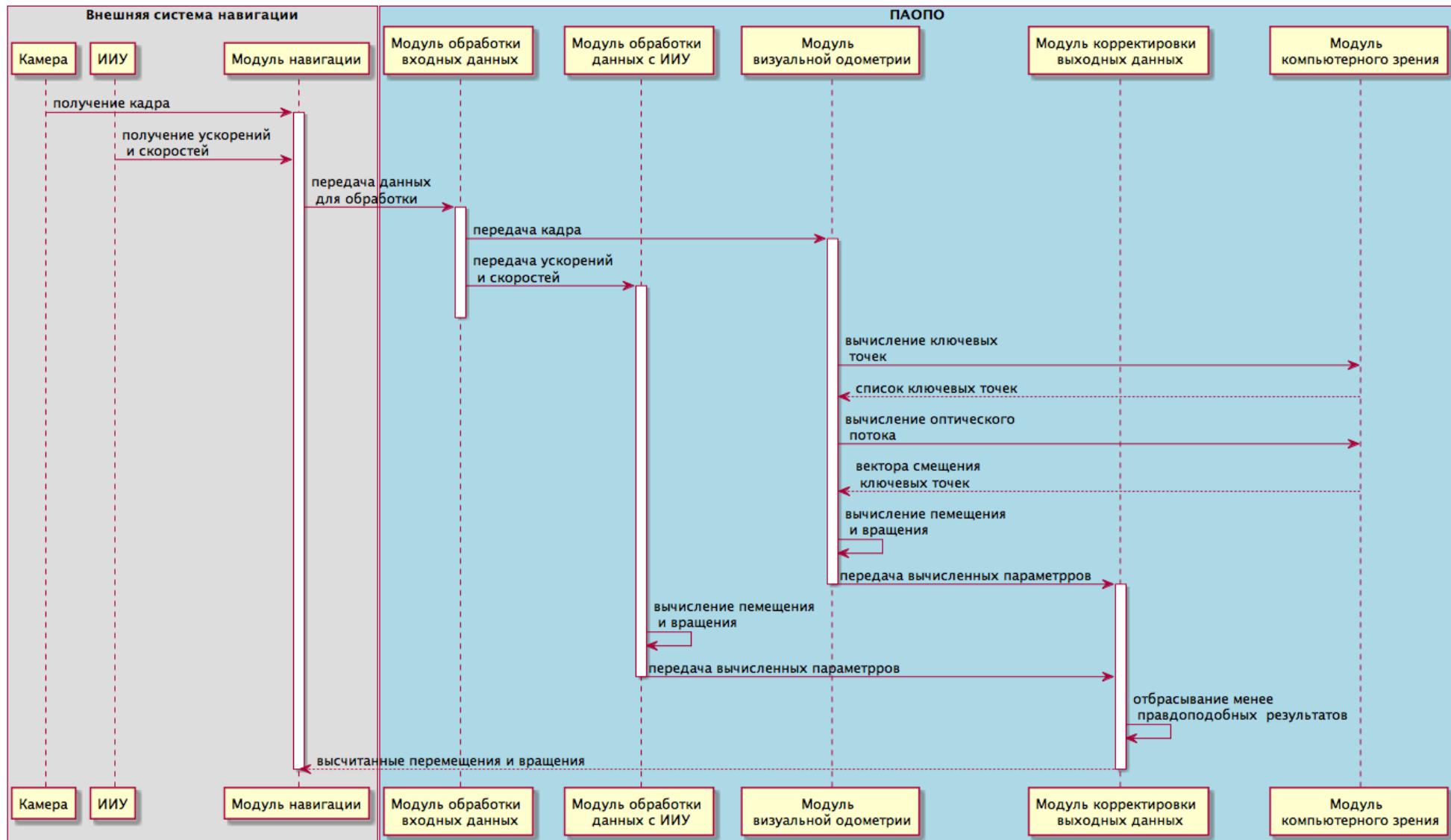


Рисунок 22 – Диаграмма последовательности взаимодействия компонентов.

## **2.2 Диаграмма взаимодействия компонентов**

В ходе работы подсистемы происходит взаимодействие ее модулей между собой и с внешними системами. Диаграмма последовательности такого взаимодействия показана на рисунке 22.

## **2.3 Диаграмма классов подсистемы**

При реализации прототипа рассматриваемой подсистемы был написан программный продукт на языке Java в программном пакете JetBrains IDEA. В качестве модуля компьютерного зрения использовалась библиотека OpenCV (Open Computer Vision). В результате был получен программный продукт состоящий из нескольких классов, связанных между собой. Получившаяся диаграмма классов представлена на рисунке 23.

## **2.4 Выбор языка программирования**

В настоящее время существует большое число языков программирования высокого уровня. Каждый из них обладает своими преимуществами и недостатками. Так как одной из целей создания ПАОПО является создание удобного в интеграции компонента, то выбор языка программирования является важным этапом при реализации подсистемы.

При этом на выбор языка программирования накладывается ограничение из-за необходимости поддержки библиотеки компьютерного зрения OpenCV. Согласно документации [14], OpenCV поддерживает следующие языки программирования:

- C++;
- C#;
- Java;
- Python;

Проведем выбор среди этих языков программирования методом взвешенных сумм по следующим критериям:

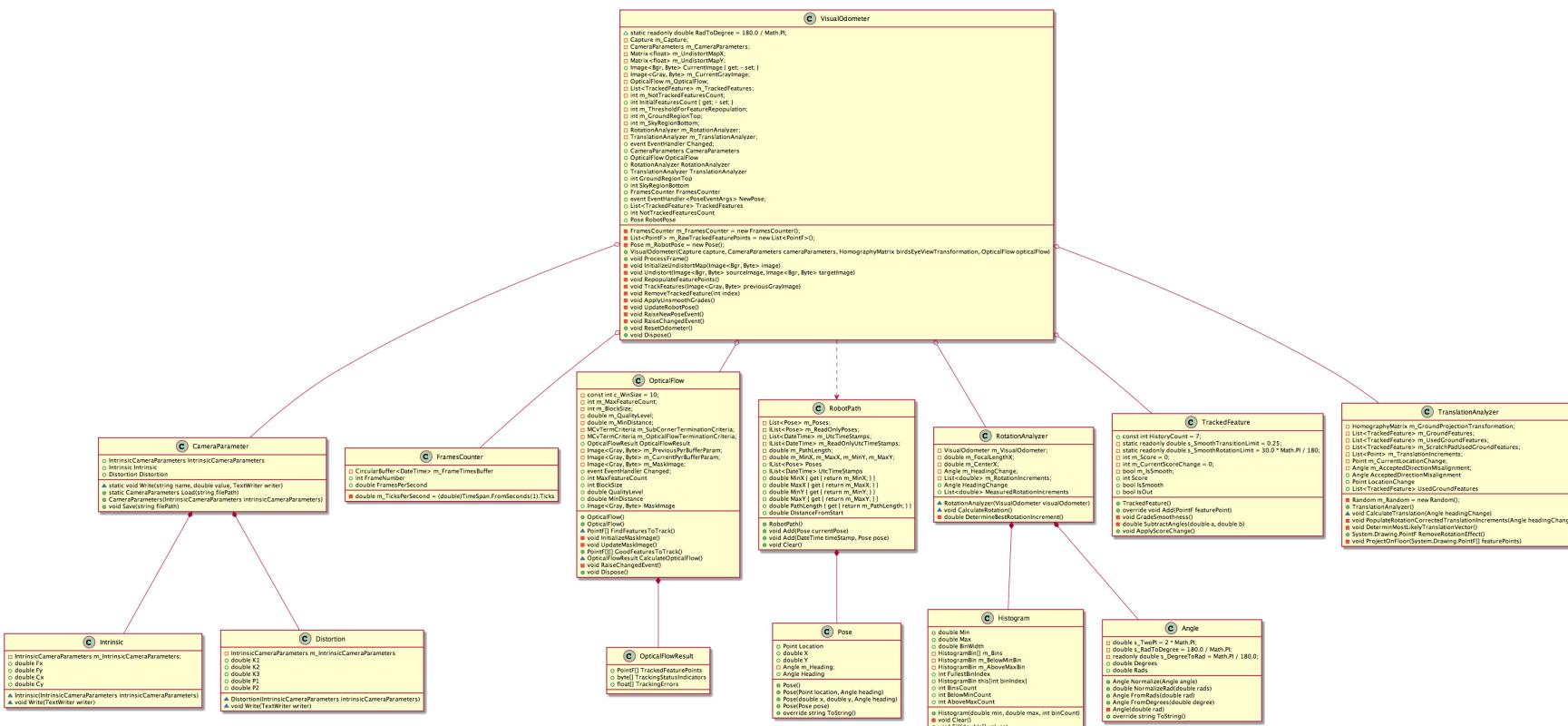


Рисунок 23 – Диаграмма классов прототипа подсистемы.

- кроссплатформенность;
- производительность;
- скорость разработки;

Сравнение возможных языков по выбранным критериям приведено в таблице 9.

Таблица 9 – Сравнение языков программирования. Лингвистические оценки.

Критерий	C++	C#	Java	Python
Кроссплатформенность	Да	Нет	Да	Да
Производительность	Высокая	Средняя	Средняя	Средняя
Скорость разработки	Низкая	Высокая	Высокая	Высокая

Переведем лингвистические оценки в числовые.

Таблица 10 – Сравнение языков программирования. Числовые оценки.

Критерий	C++	C#	Java	Python
Кроссплатформенность	1	0,4	1	1
Производительность	1	0,6	0,8	0,6
Скорость разработки	0,3	1	1	0.7

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

- кроссплатформенность -  $2x$ ;
- производительность -  $3x$ ;
- скорость разработки -  $1x$ ;

Сумма весовых коэффициентов по всем критериям качества равна 1. тогда:

$$3x + 2x + x = 6x \Rightarrow x = 0,16667$$

Таким образом, можно составить сравнительную таблицу, содержащую количественные и качественные оценки и подсчитать итог, методом взвешенной суммы. Итоговое сравнение с учетом весового коэффициента и количественных оценок приведены в таблице 11.

Таблица 11 – Сравнение языков программирования. Весовой коэффициент и итоговые оценки.

Критерий	Весовой коэффиц.	C++	C#	Java	Python
Кроссплатформенность	0,33	1	0,4	1	1
Производительность	0,5	1	0,6	0,8	0,6
Скорость разработки	0,16667	0,3	1	1	0,7
<b>Итог</b>		0,88	0,599	0,89	0,746

### 3 Исследовательская часть

#### 3.1 Обоснование области исследования

**Оптический поток (*Optical flow*)** – технология, использующаяся в различных областях компьютерного зрения для определения сдвигов, сегментации, выделения объектов, компрессии видео и прочего. В специализированной литературе и глобальной сети Internet много вводных статей в эту технологию, в которых рассмотрены основные шаги для реализации оптического потока в своих проектах. Однако, при попытке реализовать оптический поток самостоятельно на основе полученной информации, такие реализации работают очень плохо и сбоят при определении сдвигов уже порядка 1-2 пикселей.

В таких случаях следует обратиться к готовым реализациям этих методов. В настоящий момент наиболее развитой библиотекой компьютерного зрения является OpenCV[18]. Но в этой библиотеке реализовано множество способов вычисления оптического потока, у каждого из которых несколько параметров, которые как-то оказывают влияние на качество и скорость работы алгоритмов.

В данной работе оптический поток играет ключевую роль, а время его вычисления занимает более половины времени работы всей системы. В связи с чем, в исследовательской части будут рассмотрены реализованные в OpenCV методы вычисления оптического потока, исследовано влияние их параметров на скорость работы алгоритмов, а так же оценено время работы каждого из методов при разных входных данных, приближенных к условиям работы разрабатываемой системы.

#### 3.2 Описание исследуемых методов

##### 3.2.1 метод Лукаса-Канаде

Метод Лукаса-Канаде является стандартным подходом к вычислению оптического потока, так как обладает достаточно низкой ресурсоемкостью и временем выполнения. В основном это достигается за счет вычисления

не плотного оптического потока, а выборочного, для определенных точек изображения. Подробно этот алгоритм был рассмотрен в пункте 1.2.1.4.

### 3.2.2 метод Horn–Schunck

Метод Horn–Schunck носит несколько более глобальный характер, чем метод Лукаса–Канаде. Он опирается на предположение о том, что на всем изображении оптический поток будет достаточно гладким.

Этот алгоритм был реализован в первых версиях OpenCV, но впоследствии от него отказались по причине его плохой приспособленности к реальным видеопотокам. Данный метод упомянут только для ознакомления, в дальнейшем он не будет рассматриваться.

### 3.2.3 метод Farneback

В данном методе предлагается аппроксимировать изменение интенсивности в окрестности с помощью квадратичной формы:  $I = xAx + bx + c$  с симметричной матрицей  $A$  (по сути, рассматривая разложение по Тейлору до первого члена, мы брали линейную аппроксимацию  $I = bx + c$ . используя указанное разложение мы увеличиваем точность аппроксимации.) Если изображение сдвинулось в пределах этой окрестности, то  $I_2(x) = I_1(x - d)$ , подставив в квадратичное разложение и раскрывая скобки, получаем:

$$A_2 = A_1$$

$$b_2 = b_1 - 2A_1d$$

$$c_2 = d^T A_1 d - b_1^T d + c_1$$

Вычисляя значения  $A, b, c$  на обеих картинках, получим избыточную систему относительно  $d$ . Более того, приближенное значение  $d$  можно получить из второго уравнения:

$$d = -\frac{1}{2} \cdot A^{-1} \cdot cdot(b_2 - b_1)$$

### **3.2.4 метод SimpleFlow**

В основе метода SimpleFlow лежит следующая идея: если все равно нет возможности определять сдвиг больше чем размер окна, по которому мы искали производные, то можно просто в окне найти наиболее похожую точку. А для разрешения неоднозначность и для компенсации шумов учитывать, что поток непрерывный и в окрестности данной точки все точки имеют почти одинаковый сдвиг. Возникающие трудности с размером окна возможно решить за счет multi-scaling'a<sup>1</sup>.

### **3.2.5 метод Dual TV L1**

В ходе написания данной работы, 25 апреля 2014 года, была выпущена новая версия OpenCV 2.4.9, в которой был реализован еще один метод вычисления оптического потока - Dual TV L1. К сожалению к этому моменту эксперимент был завершен и исследование еще этого метода не представлялось возможным.

## **3.3 Проведение эксперимента**

Целью эксперимента было выявление зависимостей времени вычисления оптического потока разными методами от размера кадров и различных параметров этих методов.

### **3.3.1 Методика проведения эксперимента**

Эксперимент проводился следующим образом. Для анализа производительности каждым методом анализировался видеофайл длительностью 10 секунд и с частотой кадров - 30 кадров/сек. Итого оптический поток был высчитан для 299 кадров (для первого кадра не вычисляется). Размер кадра изменялся от 160\*90 до 1280\*720 пикселей с увеличением кадра в 2 раза на каждом teste.

---

<sup>1</sup>итеративное вычисление оптического потока для уменьшенных изображений с целью определения больших сдвигов.

### 3.3.2 Исходный код программы

Для проведения эксперимента была написана программа на языке Java с использованием библиотеки JavaOpenCV.

```
1 public Mat onCameraFrame(CvCameraViewFrame inputFrame) {  
2  
3     mGray = inputFrame.gray();  
4     Mat resizeImg = new Mat();  
5  
6     if (mPrevGray != null && !mPrevGray.empty()) {  
7  
8         MatOfKeyPoint mKeyPoint = new MatOfKeyPoint();  
9         FeatureDetector.create(FeatureDetector.HARRIS).detect(mPrevGray, mKeyPoint);  
10  
11         ArrayList<Point> temp = new ArrayList<Point>();  
12         List<KeyPoint> keyPointsList = mKeyPoint.toList();  
13         for (int i = 0; i < keyPointsList.size(); i++)  
14         {  
15             temp.add(keyPointsList.get(i).pt);  
16         }  
17         MatOfPoint2f needs = new MatOfPoint2f();  
18         needs.fromList(temp);  
19         MatOfPoint2f result = new MatOfPoint2f();  
20         MatOfByte status = new MatOfByte();  
21         MatOfFloat err = new MatOfFloat();  
22  
23         calcOpticalFlowPyrLK(mPrevGray, mGray, needs,  
24 , result, status, err);
```

```
25      }
26
27      mPrevGray = mGray;
28      count++;
29
30 return mGray;
31 }
```

### 3.3.3 Тестовый стенд

Сначала тестирование проводилось на настольном персональном компьютере со следующими характеристиками:

- процессор Intel Core i7 с тактовой частотой 3100МГц;
- 4ГБ оперативной памяти.

Однако при такой конфигурации оборудования отличия во времени выполнения на разных размерах кадра были незначительными, поэтому эксперимент был проведен на смартфоне LG Nexus 4 со следующими характеристиками:

- процессор Nvidia с тактовой частотой 1500МГц;
- 2ГБ оперативной памяти.

### 3.3.4 Результаты эксперимента

Отлический поток высчитывался 3-мя методами: SimpleFlow, Farneback и Lucas-Kanade. Все эти три метода реализованы в библиотеке компьютерного зрения OpenCV и вначале вызывались со стандартными параметрами. Замерялось время вычисления ОП для всей видеопоследовательности. Результаты эксперимента представлены на рисунке 24.

Из графика видно, что метод Лукаса-Канаде является самым быстрым, что обусловлено тем, что этим методом вычисляется лишь выборочный поток, а не плотный.

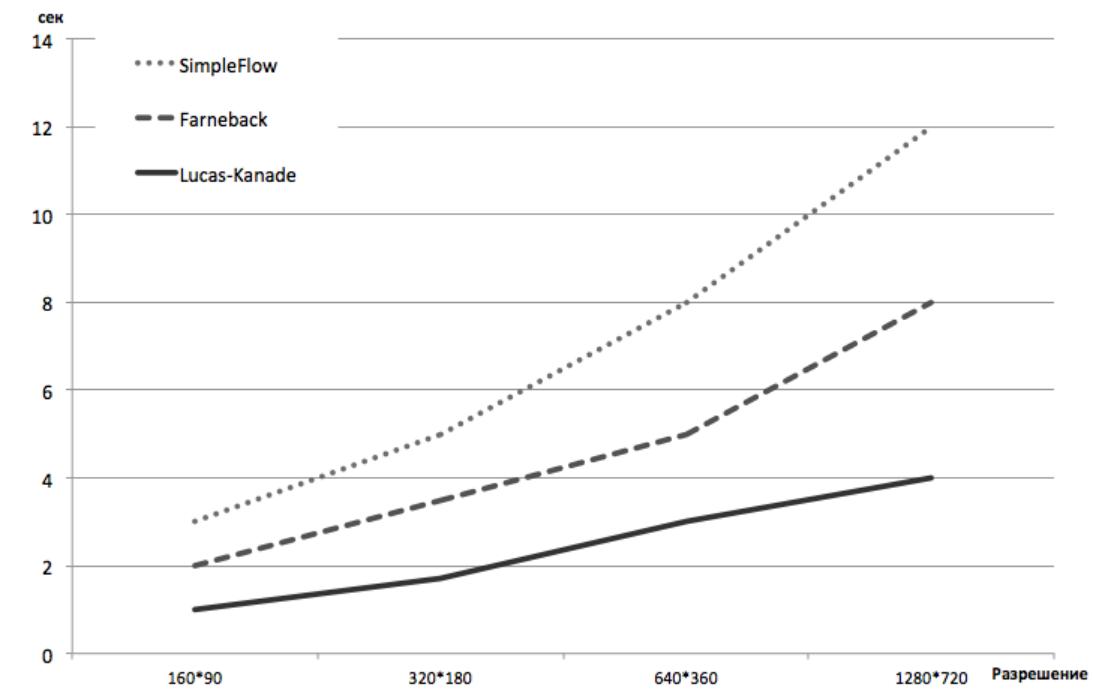


Рисунок 24 – Сравнение времени выполнения методов вычисления оптического потока

Тем не менее, методы обладают рядом параметров, которые могут позволить увеличить их производительность. В связи с этим было проведено исследование их влияния на общее время вычисления оптического потока.

### 3.3.4.1 Farneback

Согласно документации OpenCV метод Farneback имеет следующие параметры.

- *prev* – первое изображение.
- *next* – второе изображение.
- *flow* – вычисленный оптический поток в виде векторов.
- *pyr-scale* – параметр, характеризующий коэф. уменьшения изображений при построении пирамиды. Стандартное значение - 0,5.
- *levels* – число уровней в пирамиде. Стандартное значение - 3.
- *winsize* – средний размер окна. Стандартное значение - 15.

- *iterations* – число итераций на каждом уровне пирамиды. Стандартное значение - 3.
- *poly-n* – число пикселей, используемых для идентификации ключевых точек. Стандартное значение - 5.
- *poly-sigma* – стандартное отклонение Гаусса, используемое для сглаживания производных. Стандартное значение - 1,1.

Очевидно, что первые три параметра изменять нельзя, кроме разме-ра самих кадров, что мы и так делаем. На рисунке 25 показано влияние других параметров на время вычисления оптического потока.

Из графиков видно, что снизить время выполнения можно путем уменьшения уровней в пирамиде, однако такой подход снижает точность вычисления оптического потока. К сожалению, выразить точность вычисления в измеримой величине сложно. Такая же ситуация с числом итераций *interation* – снижение их числа снижает время выполнения, но качество вычисления потока снижается.

### 3.3.4.2 SimpleFlow

Согласно документации OpenCV метод SimpleFlow имеет следующие параметры.

- *prev* – первое изображение.
- *next* – второе изображение.
- *flow* – вычисленный оптический поток в виде векторов.
- *layers* – число уровней при построении пирамиды. Стандартное значение - 3.
- *averaging-block-size* – размер окна, по которому происходит поиск пик-селей. Стандартное значение - 2.
- *max-flow* – максимальный сдвиг для всех уровней пирамиды. Стандарт-ное значение - 4.

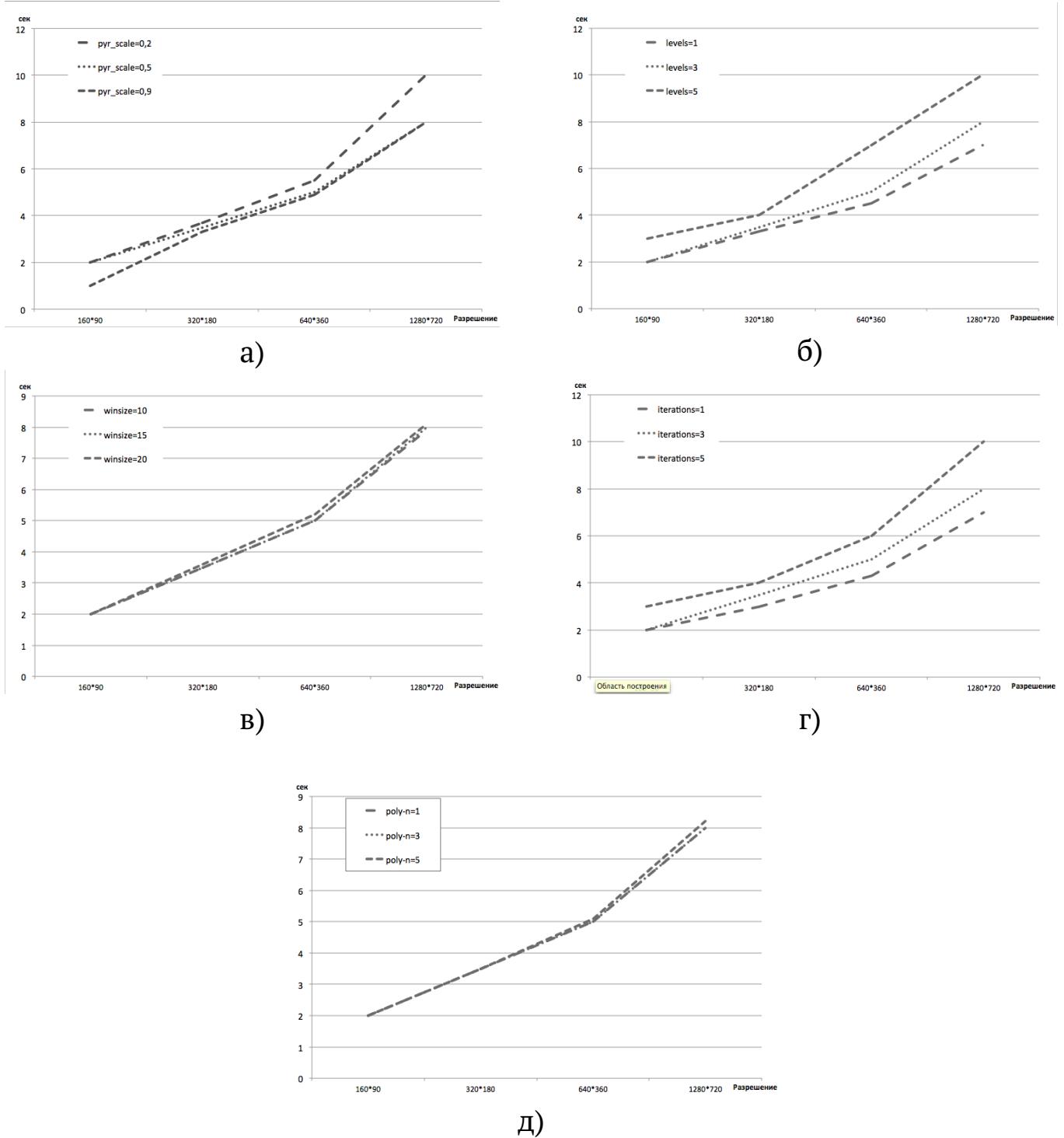


Рисунок 25 – Влияние значения параметров на время работы алгоритма:  
а)pyr-scale; б) levels; в) winsize ; г) interation; д) poly-n

Так же очевидно, что первые три параметра являются опорными и не подлежат изменению в рамках эксперимента. Влияние последних трех параметров на время вычисления оптического потока методом SimpleFlow показано на рисунке 26

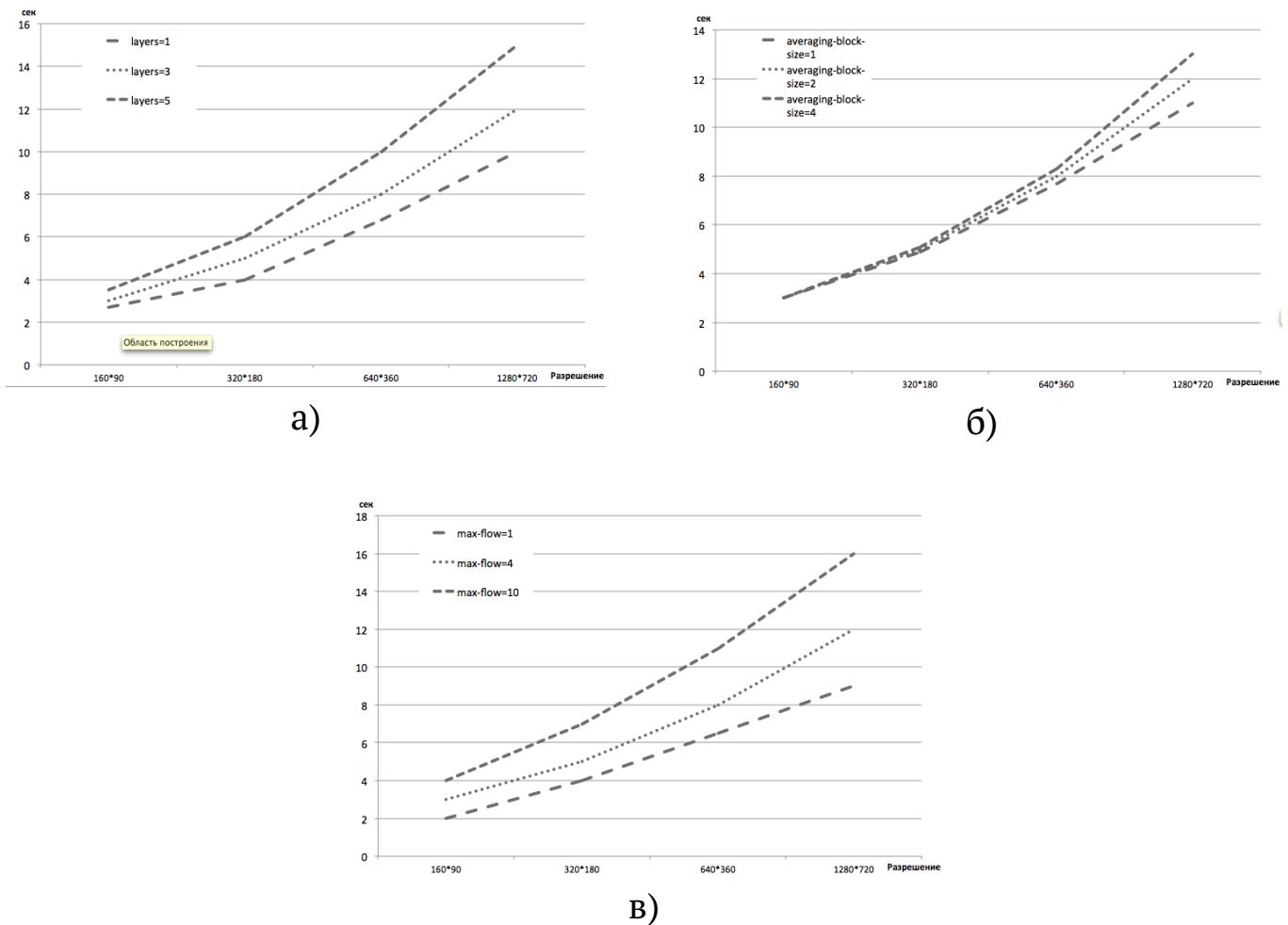


Рисунок 26 – Влияние значения параметров на время работы алгоритма:  
а) layers; б) averaging-block-size; в) max-flow;

Из графиков видно, что наибольшее влияние оказывает параметр max-flow, но так как он ограничивает значения максимального оптического потока, то для нахождения всех возможных сдвигов в ходе быстрого движения в кадре, этот параметр должен быть близок к  $\max(\text{высота кадра}, \text{ширина кадра})$ . Таким образом, уменьшение данного параметра до 1, снижает время вычисления, но делает работу алгоритма бесмысленной, так как сдвиги более чем на 1 пиксель не будут найдены.

### 3.4 Выводы к исследовательской части

В ходе проведения экспериментов было установлено следующее:

- Алгоритм Лукаса-Канаде работает быстрее алгоритмов SimpleFlow и Farneback. Причем разница более заметна при больших разрешениях кадра.
- Алгоритмы SimpleFlow и Farneback имеют несколько параметров, оказывающих влияние на время их выполнения, однако при уменьшении времени выполнения снижается и качество вычисления результирующего оптического потока.
- При необходимости высчитывать оптический поток в реальных условиях, самым приемлемым является алгоритм Лукаса-Канаде, которые показывает наименьшее время выполнения при приемлемом качестве. Такое преимущество возникает из-за вычисления не плотного, а выборочного оптического потока.

## **4 Экономическая часть**

### **4.1 Обоснование сметы затрат на разработку программного продукта ПАОПО**

Процесс разработки сложного программного продукта сопровождается необходимостью решения многих экономических проблем. Одна из важных экономических проблем – определение стоимости программного продукта (ПП), т.е. сметной стоимости затрат на его разработку.

Затраты на разработку программного продукта могут быть представлены в виде сметы затрат, включающей в себя следующие статьи:

- расходные материалы;
- затраты на оборудование;
- затраты на оплату труда;
- накладные расходы;
- услуги сторонних организаций;
- прочие расходы;

Расчет затрат на разработку данного программного продукта проводился для уровня цен и окладов на 22.04.2014г.

#### **4.1.1 Расчет затрат на расходные материалы**

В статье учитываются суммарные затраты на расходные материалы, приобретаемые для разработки данного программного продукта (ПП), которые указаны в Таблице 12.

Получаем, что затраты на расходные материалы составляют СМ=24 242 руб.

Таблица 12 – Стоимости расходных материалов и инструментов

Наименование	Кол-во	Цена
Win Home Basic 7 SP1 32-bit Russian	2	2 464 руб
Visual Studio Professional 2012	1	13 998 руб
IntelliJ IDEA 13	1	7 500 руб
канцелярские товары		
писчая бумага А4 (пачка)	1	140 руб
ручки, карандаши, ластики		100 руб
CD – RW диск	1	40 руб
Итого: 24 242 руб		

#### 4.1.2 Расчет затрат на оборудование

В статье учитываются суммарные затраты на использование оборудования.

$$C_{\text{ЭВМ}} = \frac{\Pi_{\text{ЭВМ}} \cdot T_{\text{ЭВМ}}}{T_{\text{AMP}}} = \frac{12000 \cdot 3}{5 \cdot 12} = 600 \text{ руб}$$

где,

$C_{\text{ЭВМ}}$  — затраты на использование (аренду) ПЭВМ для разработки программного продукта

$\Pi_{\text{ЭВМ}}$  — покупная цена вычислительной техники:  $\Pi_{\text{ЭВМ}} = 12 000$  руб

$T_{\text{ЭВМ}}$  — время использования ПЭВМ для разработки данного программного продукта в месяцах (3 месяца)

$T_{\text{AMP}}$  — срок амортизации вычислительной техники, составляет 5 лет.

Тогда  $T_{\text{AMP}} = 5$  лет =  $5 \cdot 12 = 60$  месяцев.

Затраты на ремонт вычислительной техники составляют 5% от стоимости ее использования и равны:

$$C_{\text{рем}} = 0,05 \cdot C_{\text{ЭВМ}} = 30 \text{ руб}$$

Получаем, что затраты на оборудование с учетом его ремонта составляют:  $C_{\text{ОБ}} = C_{\text{ЭВМ}} + C_{\text{РЕМ}} = 600 + 30 = 630$  руб.

## 4.2 Определение трудоемкости выполнения проекта

Трудоемкость разработки проекта по каждому участнику может быть определена как сумма величин трудоемкости выполнения участниками отдельных стадий.

В соответствии с ГОСТ 19.102-94 «Стадии разработки» процесс разработки ПОАПО разбивается на пять стадий: разработка ТЗ, эскизное проектирование, техническое проектирование, рабочее проектирование и внедрение. Этот ГОСТ допускает в технико-обоснованных случаях исключать стадии эскизного и технического проектов, то есть объединять техническое и рабочее проектирование. Трудоемкость каждого этапа указывается в часах и приведена в Таблице 13.

Для определения трудоемкости разработки проекта по каждому участнику в человеко-днях, используем следующую формулу:

$$T_{\text{рд}} = T_{\text{час}} / t_{\text{рд}},$$

где  $T_{\text{час}}$  – время на разработку в часах,  $t_{\text{рд}}$  – коэффициент, показывающий количество рабочих часов в одном дне. Для дальнейших расчетов примем  $t_{\text{рд}} = 8$  час

Для аналитика  $T_{\text{рд}} = T_{\text{час}} / t_{\text{рд}} = 480 / 8 = 60$  дней.

Для разработчика  $T_{\text{рд}} = T_{\text{час}} / t_{\text{рд}} = 300 / 8 = 38$  дней.

Или, суммарно, – 98 рабочих дней.

Для определения времени реализации проекта требуется перевести рабочие дни в календарные дни (КД). Для перевода используется следующая формула:

$$T_{\text{кд}} = \frac{T_{\text{РД}} \cdot (1 + d)}{g},$$

где  $d$  – доля дополнительных работ, порученных другой группе работников попутно с основной работой (от 0,1 до 0,3). В нашем случае проект ведётся самостоятельно,  $d = 0$ ,  $g$  – коэффициент перевода (в зависимости от выходных и праздничных дней) – 0,73.

Для аналитика:  $T_{\text{кд}} = T_{\text{РД}} \cdot (1 + d) / g = 60 / 0,73 = 83$  календарных дня.

Для разработчика:  $T_{\text{кд}} = T_{\text{РД}} \cdot (1 + d) / g = 38 / 0,73 = 52$  календарных дня.

Таблица 13 – Трудоемкость по этапам проектирования

Стадия	Этап	Трудоёмкость (часы)	
		Аналитик	Разработчик
1. Разработка ТЗ	1.1 Формулировка и уточнение задания.	20	–
	1.2. Исследование и анализ ПО.	80	–
	1.3. Разработка и утверждение ТЗ.	40	–
2. Рабочее проектирование	2.1. Техническое проектирование (разработка моделей данных и алгоритмов).	200	20
	2.2. Рабочее проектирование (кодирование и тестирование программных модулей).	20	160
	2.3. Тестовые испытания системы.	20	100
	2.4. Разработка программной документации.	60	–
3. Внедрение	3.1. Подготовка проекта к внедрению.	40	20
Итого по участникам:		480	300
Итого:			780

Или, суммарно, – 135 календарных дней.

### 4.3 Расчет затрат на оплату труда

В данную статью включается заработная плата исполнителей, непосредственно связанных с разработкой программного продукта, с учетом их должностного оклада и времени участия в разработке.

Основная заработная плата рассчитывается по формуле:

$$ЗП_{\text{осн}} = T_{\text{д}} \cdot L_{\text{ср.дн.}},$$

где  $T_{\text{д}}$  – трудоемкость, календарные дни,  $L_{\text{ср.дн.}}$  – среднедневной заработка работника.

Для определения средней заработной платы аналитика-руководителя небольшого проекта и программиста-разработчика проведен анализ основных ресурсов, предоставляющих сервисы для поиска работы и дающих возможность оценить размер компенсации труда. Такой подход позволяет оценить максимально приближенную к реальности рыночную стоимость труда.

Использованные ресурсы:

- портал поиска предложений по трудуустройству HeadHunter. Адрес: [www.hh.ru](http://www.hh.ru)
- портал поиска предложений по трудуустройству Job.ru. Адрес: [www.job.ru](http://www.job.ru).

В результате в качестве средней заработной платы аналитика-руководителя проекта было взято 50 тыс. руб., программиста – 60 тыс. руб.

Тогда среднедневной заработка находится по формуле:

$$L_{\text{ср.дн.}} = L_0/F,$$

где  $L_0$  – среднемесячная заработная плата,  $F$  – среднее количество рабочих дней в месяце.  $F$  вычисляется по следующей формуле:

$$F = \frac{\sum N_{\text{раб}}}{n} = \frac{18 + 19 + 22}{3} = 19,$$

где  $N_{\text{раб}}$  – количество рабочих дней в месяце,  $n$  – число месяцев.

В данном случае  $n = 3$ .

Тогда, для аналитика  $L_{\text{ср.дн.}} = 50000/19 = 2631$  руб. и расходы на основную зарплату составят:  $ЗП_{\text{осн}} = 2631 \cdot 60$  дней  $\approx 158000$  руб.

Тогда, для разработчика  $L_{\text{ср.дн.}} = 60000/19 = 3157$  руб. и расходы на основную зарплату составят:  $ЗП_{\text{осн}} = 3157 \cdot 38$  дней  $\approx 120000$  руб.

## **Дополнительная заработка плата.**

Расходы на дополнительную заработанную плату учитывают все выплаты непосредственно исполнителям за время не проработанное на производстве, но предусмотренное законодательством, в том числе: оплата очередных отпусков, компенсация за недоиспользованный отпуск, и др. Величина этих выплат составляет 20% от размера основной заработной платы:

$$C_{зпд} = 0,2 \cdot C_{зпо} = 0,2 \cdot 278\,000 = 55\,600 \text{ руб}$$

В результате получаем, что затраты на оплату труда составляют:

$$C_{зпп} = C_{зпо} + C_{зпд} = 278\,000 + 55\,600 = 333\,600 \text{ руб.}$$

### **4.3.1 Расчет затрат на страховые взносы**

В данной статье затрат учитываются отчисления на социальные нужды, производимые в фонды социального страхования, обязательного медицинского страхования и пенсионный фонд. Расчет производится с учетом законов, принятых с 1 января 2012 года (отдельные положения вступают в иные сроки):

Федеральный закон от 24.07.2009 №212 ФЗ (ред. от 28.12.2013) «О страховых взносах в Пенсионный фонд Российской Федерации, Фонд социального страхования Российской Федерации, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования»;

Федеральный закон от 24.07.2009 №213 ФЗ (ред. от 07.05.2013) «О внесении изменений в отдельные законодательные акты Российской Федерации и признании утратившими силу отдельных законодательных актов (положений законодательных актов) Российской Федерации в связи с принятием Федерального закона «О страховых взносах в Пенсионный фонд Российской Федерации, Фонд социального страхования Российской Федерации, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования».

С 1-го января 2006 года согласно федеральному закону РФ №158-ФЗ от 6.12.2005 года величина единого социального налога рассчитывается по формуле:

$$C_{\text{сн}} = K^{\text{сн}} \cdot C_{\text{зп}},$$

где  $K^{\text{сн}}$  – коэффициент, учитывающий социальный налог,  $C_{\text{зп}}$  – заработка плата (руб.)

Плательщиками страховых взносов являются страхователи, определяемые в соответствии с федеральными законами о конкретных видах обязательного социального страхования, к которым относятся:

1. лица, производящие выплаты и иные вознаграждения физическим лицам:

- организации;
- индивидуальные предприниматели;
- физические лица, не признаваемые индивидуальными предпринимателями;

2. индивидуальные предприниматели, адвокаты, нотариусы, занимающиеся частной практикой, и иные лица, занимающиеся в установленном законодательством Российской Федерации порядке частной практикой (далее - плательщики страховых взносов, не производящие выплаты и иные вознаграждения физическим лицам), если в федеральном законе о конкретном виде обязательного социального страхования не предусмотрено иное. (в ред. Федерального закона от 03.12.2011 №379-ФЗ). Для страхователей, перечисленных выше, предусмотрены следующие ставки:

ПФР	ФСС	ФФОМС
22%	2,9%	5,1%

Отсюда  $K^{\text{сн}} = 0,3$  и таким образом затраты на страховые взносы составляют:  $C_{\text{сн}} = 0,3 \cdot 333\,600 \text{ руб} = 100\,080 \text{ руб.}$

#### **4.3.2 Расчет затрат на услуги сторонних организаций**

В статье учитываются затраты на выполнение сторонними организациями работ, непосредственно связанных с разработкой программного продукта.

При разработке данного продукта потребовались услуги сторонних организаций по изготовлению 10-ти плакатов формата А1 и печати на принтере 300 листов РПЗ формата А4. Стоимость распечатки плакатов (СПЛ) и листов РПЗ (СЛ) соответственно рассчитываются по формулам:

$$C_{\text{ПЛ}} = 10 \cdot C_{\text{A1}} = 10 \cdot 150 = 1500 \text{ руб.},$$

где  $C_{\text{A1}}$  – стоимость распечатки одного плаката формата А1.  $C_{\text{A1}} = 150$  руб..

$$C_{\text{Л}} = 300 \cdot C_{\text{A4}} = 300 \cdot 2 = 600 \text{ руб.},$$

где  $C_{\text{A4}}$  – стоимость распечатки одного листа формата А4.  $C_{\text{A4}} = 2$  руб.

Получаем, что затраты на услуги сторонних организаций составляют  $C_{\text{изг}} = C_{\text{ПЛ}} + C_{\text{Л}} = 2100$  руб.

#### **4.3.3 Расчет затрат на накладные расходы**

В данной статье учитываются затраты на общехозяйственные расходы (это плата за здание, в котором идет разработка, его ремонт, плата за энергоресурсы), непроизводственные расходы и расходы на управление. Накладные расходы составляют 12,5% + 25% от уровня ФОТ.

$$C_{\text{НР}} = (0,125 + 0,25) \cdot (C_{\text{М}} + C_{\text{ОБ}} + C_{\text{ЗП}} + C_{\text{СН}} + C_{\text{изг}})$$

Таким образом, затраты на накладные расходы составляют:  $C_{\text{НР}} = (0,125 + 0,25) \cdot (18\ 480 + 630 + 333\ 600 + 100\ 080 + 2\ 100) = 170\ 583,75$  руб.

#### **4.3.4 Расчет прочих расходов**

Данная статья расходов учитывает налог на имущество и налог на транспортные средства. Налог на имущество в данном случае не платится, так как все имущество, включаемое в налогооблагаемую базу в соответствии с инструкцией «О порядке исчисления и уплаты в бюджет налога на

имущество предприятий», используется на нужды образования, и, следовательно, налогом на имущество не облагается.

Налог на владельцев транспортных средств не платится, в связи с отсутствием транспортных средств.

#### 4.3.5 Итог затрат для заказчика

Итог затрат для заказчика рассчитывается как сумма по всем выше-перечисленным статьям затрат и составляет:

$$\text{Ц} = 24\ 242 + 630 + 333\ 600 + 100\ 080 + 2100 + 170\ 583 = 625\ 473 \text{ руб.}$$

Смета затрат на разработку программного продукта приведена в Таблице 14.

Таблица 14 – Стоимости расходных материалов и инструментов

№п/п	Статья затрат	Сумма статьи (руб.)
1	Расходные материалы	24 242
2	Затраты на оборудование	630
3	Затраты на оплату труда	333 600
4	Услуги сторонних организаций	2 100
5	Накладные расходы	170 583
6	Прочие расходы	-
7	Цена	631 315

### 4.4 Основные сметы затрат на тестирование, внедрение и эксплуатацию системы

Далее рассчитаем затраты на внедрение и эксплуатацию программного продукта ПАОПО.

#### 4.4.1 Тестирование

Для тестирования разрабатываемой подсистемы необходимо выполнить следующие пункты:

- сопрячь ПАОПО с видеокамерой;
- сопрячь ПАОПО с инерциальными измерительными приборами;
- провести тестирование ПАОПО, фиксируя перемещения с помощью GPS-трекера;
- наложить полученные с помощью ПАОПО данных на данные, полученные с помощью GPS-трекера.

Таким образом необходимо закупить следующее оборудование:

- видеокамера с разрешением 720\*1080;
- трехосевой гироскоп;
- трехосевой акселерометр;
- GPS-трекер;

Данное оборудование установлено во всех современных смартфонах, работающих под управлением ОС Android, стоимость которых в настоящее время составляет 6 000 рублей и более. Например, телефон HTC Sensation, удовлетворяющий всем требованиям стоит 6200 рублей.

Для проведения сопряжения полученных данных с ПАОПО необходимо написать простое приложение, фиксирующее все данные в нужном формате для анализа разрабатываемой подсистемы, разработка которого займет до двух рабочих дней Android-разработчика.

Согласно приведенным выше данным стоимость одного дня разработчика составляет 3 157 рублей. С учетом дополнительных расходов и отчислений в страховые фонды затраты на создание приложения составят:

$$31571, 2 \cdot 1, 3 \cdot 2 = 9\ 849 \text{ руб}$$

Таким образом затраты на тестирование составят 16 149 рублей.

#### **4.4.2 Внедрение и эксплуатация**

В рамках создания ПАОПО не предусмотрены внедрение и эксплуатация, так как подсистема может работать лишь в составе другой автономной системы.

ПАОПО является кроссплатформенной, что позволяет в кратчайшие сроки интегрировать ее в любую другую систему. При этом разработчику этой системы необходимо согласовать вход и выход ПАОПО, корректно предоставляя и получая данные в/из подсистемы. Суммарные затраты на проведение интеграции могут занимать до двух рабочих дней разработчика. Согласно приведенным выше данным стоимость этой работы составит не более 6 314 рублей.

Однако данные затраты не относятся к стоимости создания ПАОПО, а перекладываются на ее потребителей.

#### **4.5 Итого. Расходы на разработку и тестирование.**

Таблица 15 – Итого. Расходы на внедрение и эксплуатацию системы.

№п/п	Статья затрат	Сумма статьи (руб.)
1	Расходные материалы	24 242
2	Затраты на оборудование	630
3	Затраты на оплату труда	333 600
4	Услуги сторонних организаций	2 100
5	Накладные расходы	170 583
6	Прочие расходы	-
7	Тестирование	16 149
		<b>Итого: 647 646</b>

## **5 Промышленная экология и безопасность**

### **5.1 Анализ опасных и вредных факторов, возникающих при работе на ПЭВМ**

Для обеспечения комфортных условий труда сотрудника организации, рабочее место которого включает ПЭВМ, необходимо выполнение следующие требования[19]:

1. требования к ПЭВМ, установленные в СанПин 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»;
2. требования к помещениям для работы с ПЭВМ;
3. требования к организации и оборудованию рабочих мест с ПЭВМ;
4. требования к микроклимату помещения, установленные в СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»;
5. требования к освещению на рабочих местах, оборудованных ПЭВМ
6. требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ;
7. требования к режиму труда и отдыха.

#### **5.1.1 Помещение для работы с ПЭВМ**

Ниже приведены основные требования, предъявляемые к помещениям, предназначенным для работы сотрудников с ПЭВМ.

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке.

Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка – 0,7 - 0,8; для стен – 0,5 - 0,6; для пола – 0,3 - 0,5.

Полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

## **5.1.2 Рабочее место оператора и положение за рабочим местом**

### **5.1.2.1 Общие требования**

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов не менее 1,2 м.

Рабочие места с ПЭВМ в помещениях с источниками вредных производственных факторов должны размещаться в изолированных кабинах с организованным воздухообменом.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками 1,5 – 2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы.

При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 – 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

### **5.1.2.2 Требования к значениям параметров рабочего места**

Значения параметров рабочего стола приведены в таблице 16.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого пара-

Таблица 16 – Значения параметров рабочего стола

Параметры рабочего стола	Значения, мм
Высота рабочей поверхности	регулировка 680-800, или фиксировано 725
Ширина рабочей поверхности	1200
Глубина рабочей поверхности	800
Высота пространства для ног	750
Ширина пространства для ног	900
Глубина пространства для ног на уровне колен	900
Глубина пространства для ног на уровне вытянутых ног	900

метра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Конструкция рабочего стула должна обеспечивать поверхность сиденья с закругленным передним краем, а также соответствие значений параметров приведённым в таблице 17.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100 – 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной стоечницы.

### 5.1.2.3 Микроклимат

Оптимальные микроклиматические условия установлены по критериям оптимального теплового и функционального состояния человека. Они обеспечивают общее и локальное ощущение теплового комфорта в течение 8-часовой рабочей смены при минимальном напряжении механиз-

Таблица 17 – Значения параметров рабочего стула (кресла)

<b>Параметры рабочего стула (кресла)</b>	<b>Значения</b>
Ширина и глубина поверхности сиденья	не менее 400 мм
Высота поверхности сиденья	400 – 550 мм
Угол наклона поверхности сиденья вперед (назад)	до 15 (5) градусов
Высота опорной поверхности спинки	280 – 320 мм
Ширина опорной поверхности спинки	не менее 380 мм
Радиус кривизны горизонтальной плоскости спинки	400 мм
Угол наклона спинки в вертикальной плоскости	± 30 градусов
Расстояние спинки от переднего края сиденья	260 – 400 мм
Длина подлокотников	не менее 250 мм
Ширина подлокотников	50 – 70 мм
Высота подлокотников над сиденьем	200 – 260 мм
Внутреннее расстояние между подлокотниками	350 – 500 мм

мов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

В помещениях, в которых работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата.

В санитарных нормах СанПиН 2.2.2/2.4.1340-03 установлены величины параметров микроклимата, создающие комфортные условия[20]. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (значительные или незначительные тепловыделения). В данном дипломном проекте рассматриваются условия труда пользователей ПЭВМ, которые относятся к категории Ia, (интенсивность энергозатрат до 120 ккал/ч (до 139 Вт), работа,

производимая сидя и сопровождающиеся незначительным физическим напряжением). Величины микроклиматических параметров приведены в таблице 18.

Таблица 18 – Оптимальные параметры микроклимата в помещении

Период года	Категория работ по уровню	Температура воздуха, °C	Температура поверхности, °C	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	22-24	21-25	60-40	0,1
Теплый	Ia (до 139)	23-25	22-26	60-40	0,1

Перепады температуры воздуха по высоте и по горизонтали, а также изменения температуры воздуха в течение смены при обеспечении оптимальных величин микроклимата на рабочих местах не должны превышать 2 °C.

#### 5.1.2.4 Аэроионный состав воздуха

В связи с тем, что разработка данного дипломного проекта производилась в помещении, в отделке и (или) меблировке которых используются синтетические материалы или покрытия, способные накапливать электростатический заряд; а так же - в которых эксплуатируется оборудование, способное создавать электростатические поля, включая видеодисплейные терминалы и прочие виды оргтехники, то необходимо учесть действующие «Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений» (СанПиН 2.2.4.1294-03), а именно:

1. Нормируемые показатели аэроионного состава воздуха
  - 1.1. Аэроионный состав воздуха устанавливается в зависимости от процессов ионизации и деионизации.
  - 1.2. Нормируемыми показателями аэроионного состава воздуха производственных и общественных помещений являются:

- концентрации аэроионов (минимально допустимая и максимально допустимая) обеих полярностей  $p^+$ ,  $p^-$ , определяемые как количество аэроионов в одном кубическом сантиметре воздуха ( $\text{ион}/\text{см}^3$ );
- коэффициент униполярности  $Y$  (минимально допустимый и максимально допустимый) определяемый как отношение концентрации аэроионов положительной полярности к концентрации аэроионов отрицательной полярности.

1.3. Минимально и максимально допустимые значения нормируемых показателей определяют диапазоны концентраций аэроионов обеих полярностей и коэффициента униполярности, отклонения от которых могут привести к неблагоприятным последствиям для здоровья человека.

1.4. Значения нормируемых показателей концентраций аэроионов и коэффициента униполярности приведены в таблице 19.

Таблица 19 – Нормируемые показатели аэроионов и коэффициенты униполярности

Нормируемые показатели	Концентрация $n^+$ ( $\text{ион}/\text{см}^3$ )	Концентрация $n^-$ ( $\text{ион}/\text{см}^3$ )	Коэффициент униполярности $Y$
Минимально допустимые	$n^+ \geq 400$	$n^- \geq 400$	$0,4 \leq Y \leq 1,0$
Максимально допустимые	$n^+ < 50000$	$n^- < 50000$	

1.5. В зонах дыхания персонала на рабочих местах, где имеются источники электростатических полей (видеодисплейные терминалы или другие виды оргтехники) допускается отсутствие аэроионов положительной полярности.

1.6. Степени вредности отклонений от означенных диапазонов концентрации аэроионов и коэффициента униполярности определяются в соответствии с классификацией условий труда по аэроионному составу воздуха.

1.7. В лечебных целях могут применяться другие показатели аэроионного состава воздуха если это предусмотрено утвержденными в установленном порядке методиками лечения или применения аэроинизаторов.

## 2. Требования к проведению контроля аэроионного состава воздуха

2.1. Контроль аэроионного состава воздуха осуществляется в следующих случаях:

- в порядке планового контроля не реже одного раза в год;
- при аттестации рабочих мест;
- при вводе в эксплуатацию оборудования либо материалов, способных создавать или накапливать электростатический заряд (включая видеодисплейные терминалы и прочие виды оргтехники);
- при оснащении рабочих мест аэроионизаторами или деионизаторами.

2.2. Проведение контроля аэроионного состава воздуха помещений следует осуществлять непосредственно на рабочих местах в зонах дыхания персонала и в соответствии с утвержденными в установленном порядке методиками контроля.

## 3. Требования к способам и средствам нормализации аэроионного состава воздуха

3.1. Если в результате контроля аэроионного состава воздуха выявляется его несоответствие нормированным показателям, рекомендуется осуществление его нормализации.

3.2. Осуществление нормализации аэроионного состава воздуха рекомендуется производить на протяжении всего времени пребывания человека на рабочем месте.

3.3. Для нормализации аэроионного состава воздуха следует применять соответствующие, прошедшие санитарно-эпидемиологическую оценку и имеющие действующее санитарно-эпидемиологическое

заключение аэроионизаторы или деионизаторы предназначенные для использования в санитарно-гигиенических целях.

3.4. Санитарно-эпидемиологическая оценка и эксплуатация аэроионизаторов и деионизаторов осуществляется в установленном порядке.

### 5.1.2.5 Шум и вибрация

На рабочем месте пользователя ПЭВМ источниками шума, как правило, разговаривающие люди, внешний шум, компьютер, принтер, вентиляционное оборудование. Основными источниками внешнего шума являются транспортные потоки на улицах и дорогах.

Допустимые уровни звукового давления, уровня звука и эквивалентные уровни звука на рабочих местах должны соответствовать требованиям СанПиН 2.2.2/2.4.1340-03.

Показатели нормируемых уровней шума для помещений офисов, рабочих помещений и кабинетов административных зданий (помещения, где могут располагаться рабочие места сотрудников – разработчиков и эксплуататоров проектируемой системы) приведены в таблице .

Основным из механических факторов производственной среды является вибрация. Она не только вредно действуют на организм, но и мешают человеку выполнять как мыслительные, так и двигательные операции.

Таблица 20 – Параметры производственного шума

Уровни звукового давления в октавных полосах со среднегеометрическими частотами										Уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц		
86 дБ	71 дБ	61 дБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50	

Шум считают в пределах нормы, когда он как по эквивалентному, так и по максимальному уровню не превышает установленные нормативные значения. Допустимые уровни шума от внешних источников в помещениях установлены при условии обеспечения нормативного воздухообмена, т.е

при отсутствии принудительной системы вентиляции или кондиционирования воздуха, должны выполняться при условии открытых форточек или иных устройств, обеспечивающих приток воздуха.

Защита от шума строительно-акустическими методами должна обеспечиваться:

- рациональным архитектурно-планировочным решением здания;
- применением ограждающих конструкций, обеспечивающих нормативную звукоизоляцию;
- применением звукопоглощающих облицовок в помещении здания;
- применением глушителей шума в системах принудительной вентиляции и кондиционирования воздуха;
- виброизоляцией инженерного и санитарно-технического оборудования зданий.

### **5.1.3 Освещение**

#### **5.1.3.1 Общие требования**

Ниже приведены основные требования, предъявляемые к освещению помещений, предназначенных для работы сотрудников с ПЭВМ.

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных помещениях, в случаях основной работы с документами, можно применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 – 500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать прямую блесткость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более  $200 \text{ кд}/\text{м}^2$ .

Следует ограничивать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать  $40 \text{ кд}/\text{м}^2$  и яркость потолка не должна превышать  $200 \text{ кд}/\text{м}^2$ .

Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более  $200 \text{ кд}/\text{м}^2$ , защитный угол светильников должен быть не менее 40 градусов.

Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 – 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ. При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пуско-регулирующими аппаратами. Допускается использование многоламповых светильников с электромагнитными пуско-регулирующими аппаратами, состоящими из равного числа опережающих и отстающих ветвей.

Применение светильников без рассеивателей и экранирующих решеток не допускается. При выполнении работы использовались лампы с экранирующими решетками.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов.

Коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1,4.

Коэффициент пульсации не должен превышать 5%.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ проводится чистка стекол оконных рам и светильников не реже двух раз в год и своевременную замену перегоревших ламп.

## 5.1.4 Расчёт освещения

### 5.1.4.1 Искусственное освещение

Так как минимальным объектом различия при работе с ПЭВМ является пиксель, размер которого составляет 0.28мм, то выполняемый вид работ считаем зрительной работой очень высокой точности[21].

Нормы для искусственного освещения приведены в таблице 21.

При работе оператора разрабатываемой подсистемы предполагается работа при среднем контрасте и светлом фоне. Таким образом, нормированная минимально-допустимая освещенность при общем освещении составляет 300 лк.

Основным методом расчета общего равномерного освещения при горизонтальной рабочей поверхности является метод светового потока (коэффициента использования). Необходимый световой поток  $\Phi_{\text{л}} \text{ (лм)}$  от одной лампы накаливания или группы ламп светильника при люминесцентных лампах рассчитывают по формуле 5.

$$\Phi_{\text{осв}} = \frac{E_{\text{н}} \cdot S \cdot z \cdot k}{N_c \cdot \eta}, \quad (5)$$

Таблица 21 – Нормы для искусственного освещения

Характеристика зрительной работы	Размер объекта различия, мм	Контраст объекта с фоном	Фон	Освещенность, лк	
				Комбинированное освещение	Общее освещение
Очень высокой точности	0,15 - 0,30	малый	темный	4000	-
		малый	средний	3000	750
			средний	2500	600
		малый	светлый	2000	500
			средний	1500	400
			большой	темный	
		средний	светлый	1000	300
			большой	750	200
			большой	средний	

где  $E_{\text{н}}$  – нормированная минимально-допустимая освещенность (лк), которая определяется нормативом (см. таблицу 21);  $S$  – площадь освещаемого помещения ( $\text{м}^2$ );  $z$  – коэффициент неравномерности освещения, который зависит от типа ламп (для ламп накаливания и дуговых ртутных ламп – 1,15, для люминесцентных ламп – 1,1);  $k$  – коэффициент запаса, согласно [22] равен 1,4;  $N_c$  – число светильников в помещении;  $\eta$  – коэффициент использования светового потока ламп, учитывающий долю общего светового потока, приходящуюся на расчетную плоскость, и зависящий от типа светильника, коэффициента отражения потолка  $\rho_{\text{п}}$  и стен  $\rho_{\text{с}}$ , высоты подвеса светильников, размеров помещения, определяемых индексом  $i$  помещения. Индекс помещения определяется по формуле 6

$$i = \frac{A \cdot B}{H_c \cdot (A + B)}, \quad (6)$$

где  $A$  и  $B$  – длина и ширина помещения, м;  $H_c$  – высота подвеса светильников над рабочей поверхностью.

Коэффициент использования светового потока ламп определяют по таблицам, приводимым в СНиП 23–05–95 в зависимости от типа светильника,  $\rho_{\text{п}}$ ,  $\rho_{\text{с}}$  и индекса  $i$ . По полученному в результате расчета по формуле

ле 5 световому потоку по ГОСТ 2239-79\* и ГОСТ 6825-91 выбирают ближайшую стандартную лампу и определяют ее необходимую мощность. Умножив электрическую мощность лампы на количество светильников  $N_c$ , можно определить электрическую мощность всего освещения помещения.

Расчет освещения от светильников с люминесцентными лампами целесообразно выполнять, предварительно задавшись типом, электрической мощностью и величиной светового потока ламп. С использованием этих данных необходимое число светильников определяют по формуле 7.

$$N_c = \frac{E_h \cdot S \cdot z \cdot k}{\Phi_{osc} \cdot N_p \cdot \eta}, \quad (7)$$

Проведем расчет и определим тип и мощность используемых ламп для помещения со следующими параметрами:

- Ширина  $a = 6\text{м}$ ;
- Длина  $b = 4\text{м}$ ;
- Высота потолков  $h = 4\text{м}$ ;
- Высота стола  $h_c = 0,8\text{м}$ ;
- Высота подвеса  $h_p = h - h_c = 3,2\text{м}$ .

Этапы расчета:

1. Вычислим индекс помещения:

$$i = \frac{6 \cdot 4}{3,2 \cdot (4 + 6)} = 0,75,$$

2. Число светильников  $N$  определим исходя из того, что будет установлено 3 ряда светильников по одному светильнику в ряду:

$$N = 3 \cdot 1 = 3\text{шт.}$$

3. Определим коэффициент использования светового потока по таблицам, приводимым в СНиП 23-05-95, как:

$$\eta = 0,4$$

- коэффициента отражения потолка  $\rho_{\text{п}} = 70\%$ ;
- коэффициента отражения стен  $\rho_{\text{с}} = 50\%$ ;

4. Определим общий необходимый световой поток при условиях:

- Коэффициент неравномерности освещения для люминесцентных ламп  $z = 1,1$ ;
- Нормированная минимально-допустимая освещенность для общего освещения равна  $E_{\text{H}} = 300 \text{ лк}$ ;
- Площадь помещения  $S = a * b = 4 * 5 = 20 \text{ м}^2$ ;
- Коэффициент запаса согласно рекомендуемым в нормативах Сан-ПиН 2.2.2/2.4.1340-03 –  $K_3 = 1,4$ ;

о

5. Определим световой поток, приходящийся на один светильник:

$$\Phi_{\text{св}} = \Phi_{\text{осв}} / N = 23\ 100 / 3 = 7\ 700 \text{ лк}$$

6. Определим световой поток, приходящийся на одну лампу (в применяемых светильниках используется две лампы):

$$\Phi_{\text{л}} = \Phi_{\text{св}} / 2 = 3\ 850 \text{ лк}$$

7. Определим тип лампы по таблице:

Таблица 22 – Световой поток наиболее распространенных люминесцентных ламп напряжением 220 В

Тип лампы	Световой поток, лм, при мощности, Вт					
	15	20	30	40	65	80
ЛДЦ	500	820	1450	2100	3050	3560
ЛД	540	920	1640	2340	3575	4070
ЛХБ	675	935	1720	2600	3820	4440
ЛБ	760	1180	2100	3000	4550	5220

Для значения светового потока  $\Phi_{\text{л}} = 3850 \text{ лм}$  подходит два типа ламп:

- ЛД, 80 Вт, 4070 лм;
- ЛБ, 65 Вт, 4550 лм.

С точки зрения минимизации расхода электроэнергии, выберем второй вариант, т.к. он более экономичный. Суммарный выигрыш в мощности ламп на указанное помещение составит:

$$\Delta = (80 - 65) \cdot 3 \cdot 2 = 120 \text{ Вт}$$

Сравнение теоретических и практических результатов представлено в таблице 23.

Таблица 23 – Сравнение теоретических и практических результатов

	<b>Теоретические (расчетные) значения</b>	<b>Практические значения</b>
Световой поток от одной лампы	3850 лм	4550 лм
Количество ламп	12	12
Световой поток от всех ламп	23 100 лм	27 300 лм

Приведенный расчет показал, что практическое значение светового потока больше теоретического на 4200 лм (15%).

### **Вывод**

В результате проведенных расчетов для рассматриваемого помещения определено:

- Общее количество светильников – 3 штук;
- Общее количество ламп – 6 штук (т.к. в каждом светильнике по две лампы);
- Тип лампы – ЛБ;
- Мощность лампы – 65 Вт;
- Общий световой поток искусственного освещения – 27 300 лм.

#### **5.1.4.2 Схема расположения светильников в помещении операторов ПЭВМ**

Светильник скомпонован из двух ламп. Светильники расположены в три ряда по одному светильнику в каждом ряду. Тип используемых ламп – ЛБ–65 (люминесцентная лампа дневного света, мощность 65 Вт).

Схема расположения светильников в помещении изображена на рис. 27

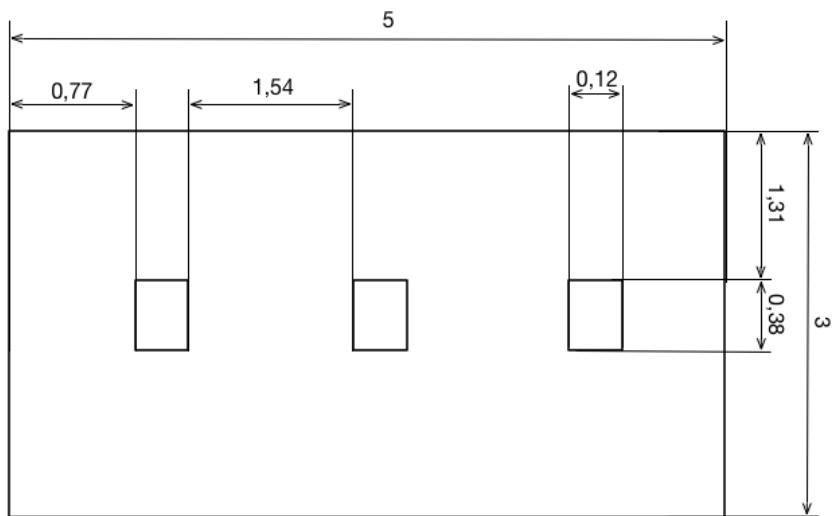


Рисунок 27 – Схема расположения светильников в помещении операторов ПЭВМ.

#### **5.2 Утилизация и списание аппаратных комплектующих**

Списание техники организация должна проводить, подтверждая факт утилизации компьютеров и оргтехники. Для организаций эта норма предписана Федеральным законом "Об отходах производства и потребления". То есть, списанная, но не утилизированная техника – это серьезное нарушение закона, не говоря уже об угрозе окружающей среде.

### **5.2.1 Стандартная процедура списания и утилизации техники**

Стандартный процесс списания техники и утилизации компьютерного оборудования включает в себя следующие шаги:

- Компания, перед которой стоит цель списания техники, создает специальную внутреннюю комиссию. Ее основной задачей будет принятие коллегиального решения о том, какую именно технику уже пора списывать.
- Решение о списании компьютеров и оргтехники данной комиссии непременно должно базироваться на экспертном заключении. Эксперт может быть как штатным сотрудником компании (обязательно иметь подтверждающие документы по образованию в сфере обслуживания/ремонта данной техники), так и привлеченным извне независимым специалистом. Обязательно потребуется акт технической экспертизы компьютеров или оргтехники, проведенной компанией-производителем, либо другой компанией, имеющей разрешение на обслуживание и ремонт данной техники. Такой акт технической экспертизы компьютеров и оборудования документально подтверждает, что техника неисправна, ее ремонт нецелесообразен и ей пора на покой. Можно списывать и утилизировать старую технику.
- Чтобы окончательно завершить списание оргтехники и компьютеров и забыть о них, придется предоставить еще и документальное подтверждение того, что он действительно был правильно утилизирован, а не продолжил уничтожать нашу экосистему, распадаясь на тяжелые металлы и ядовитые соединения.

Очевидно, что данная процедура списания компьютеров и утилизации оборудования очень долгая и сложная. Правильней было бы заключить договор на утилизацию и перепоручить утилизацию оборудования специализированной компании.

Компании предлагают услугу вывоза и утилизации компьютерной техники и оргтехники, которая поможет быстро, законно и с минимальными затратами сил и времени избавиться от старой техники.

После заключения договора, специализированная компания несет ответственность за то, чтобы провести правильное списание оргтехники и корректно выполнить вывоз и утилизацию компьютеров и прочей старой техники. При утилизации старых компьютеров происходит их разработка на фракции: металлы, пластмассы, стекло, провода, штекеры. Из одной тонны компьютерного лома получают до 200 кг меди, 480 кг железа и нержавеющей стали, 32 кг алюминия, 3 кг серебра, 1 кг золота и 300 г палладия.

Переработку промышленных отходов производят на специальных полигонах, создаваемых в соответствии с требованиями СНиП 2.01.28-85 и предназначенных для централизованного сбора обезвреживания и захоронения токсичных отходов промышленных предприятий, НИИ и учреждений.

# **Заключение**

В результате выполнения дипломного проекта решены следующие задачи:

- проведено исследование предметной области, выбраны и обоснованы критерии качества системы;
- разработана структура подсистемы, произведен анализ аналогов системы;
- проработана математическая модель;
- разработана архитектура системы;
- разработана тестовая версия подсистемы;
- произведено исследование зависимости времени вычисления оптического потока разными методами в зависимости от разных параметров;
- разработаны рекомендации по организации рабочего места разработчика подсистемы;
- рассчитаны затраты на разработку и тестирование системы;
- разработана графическая часть документации в составе 10 листов формата А1.

Разработанная подсистема является важным структурным компонентом для различных систем управления подвижными объектами.

## **Список используемых источников**

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. Авиация: Энциклопедия. М.: Большая Российская Энциклопедия, 1994.
7. 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].
8. Geiger Andreas, Ziegler Julius, Stiller Christoph. StereoScan: Dense 3d Reconstruction in Real-time // Intelligent Vehicles Symposium (IV). 2011.
9. Kitt Bernd, Geiger Andreas, Lategahn Henning. Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme // Intelligent Vehicles Symposium (IV). 2010.

10. Project Tango от Google: смартфон с 3D-сканером пространства. Коллективный блог Habrahabr. Режим доступа: <http://habrahabr.ru/post/213381/>. [Дата обращения 13.05.2014].
11. Project Tango. Project official site. Режим доступа: <https://www.google.com/atap/projecttango/>. [Дата обращения 13.05.2014].
12. Визуальное определение местоположения робота. SMP robotics. Режим доступа: <http://www.smprobotics.ru/technologies/vizualnoe-opredelenie-mestopolozheniya-robota/>. [Дата обращения 13.05.2014].
13. 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].
14. Gary Bradski Adrian Kaehler. Learning OpenCV. First Edition изд. O'Reilly, 2008. September.
15. Калибровка камеры. Свободная энциклопедия Википедия. Режим доступа: [http://en.wikipedia.org/wiki/Camera\\_resectioning](http://en.wikipedia.org/wiki/Camera_resectioning). [Дата обращения 20.05.2014].
16. Калибровка камеры на OpenCV. Разработка роботов. Режим доступа: <http://robot-develop.org/archives/2026>. [Дата обращения 05.05.2014].
17. Zhang Zhengyou. A flexible new technique for camera calibration: Tech. Rep.: : IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.
18. OpenCV. Свободная энциклопедия Википедия. Режим доступа: <http://ru.wikipedia.org/wiki/OpenCV>. [Дата обращения 22.05.2014].
19. Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы".

20. Санитарно-эпидемиологические правила и нормативы СанПин 2.2.4 1294-03 "Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений".
21. СП 52.13330.2011 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*».
22. Расчет искусственного освещения: Методическое пособие по разделу “Охрана труда” в дипломных проектах. МГТУ им.Баумана. М, 1976.

## **Приложение А Техническое задание**

## **Приложение В Графическая часть**

# Подсистема автономного определения положения объекта

## Цель разработки

Целью разработки подсистемы автономного определения перемещения объекта является создание удобного, дешевого и не требующего специализированных устройств для своей работы компонента для интеграции в подвижные автономные системы.

## Задачи, решаемые в дипломном проекте

1. изучение моделей, методик и технологий одометрии, а также связанных с ней областей;
2. анализ полученной информации;
3. выбор применяемых методов и устройств;
4. выработка спецификаций и требований к ПАОПО;
5. проектирование общей структуры системы в виде связанных модулей;
6. проектирование общей схемы взаимодействия модулей;
7. проектирование общего алгоритма функционирования подсистемы;
8. детальная разработка структуры и алгоритма работы модулей;
9. кодирование и отладка отдельных модулей;
10. отладка и тестирование подсистемы.

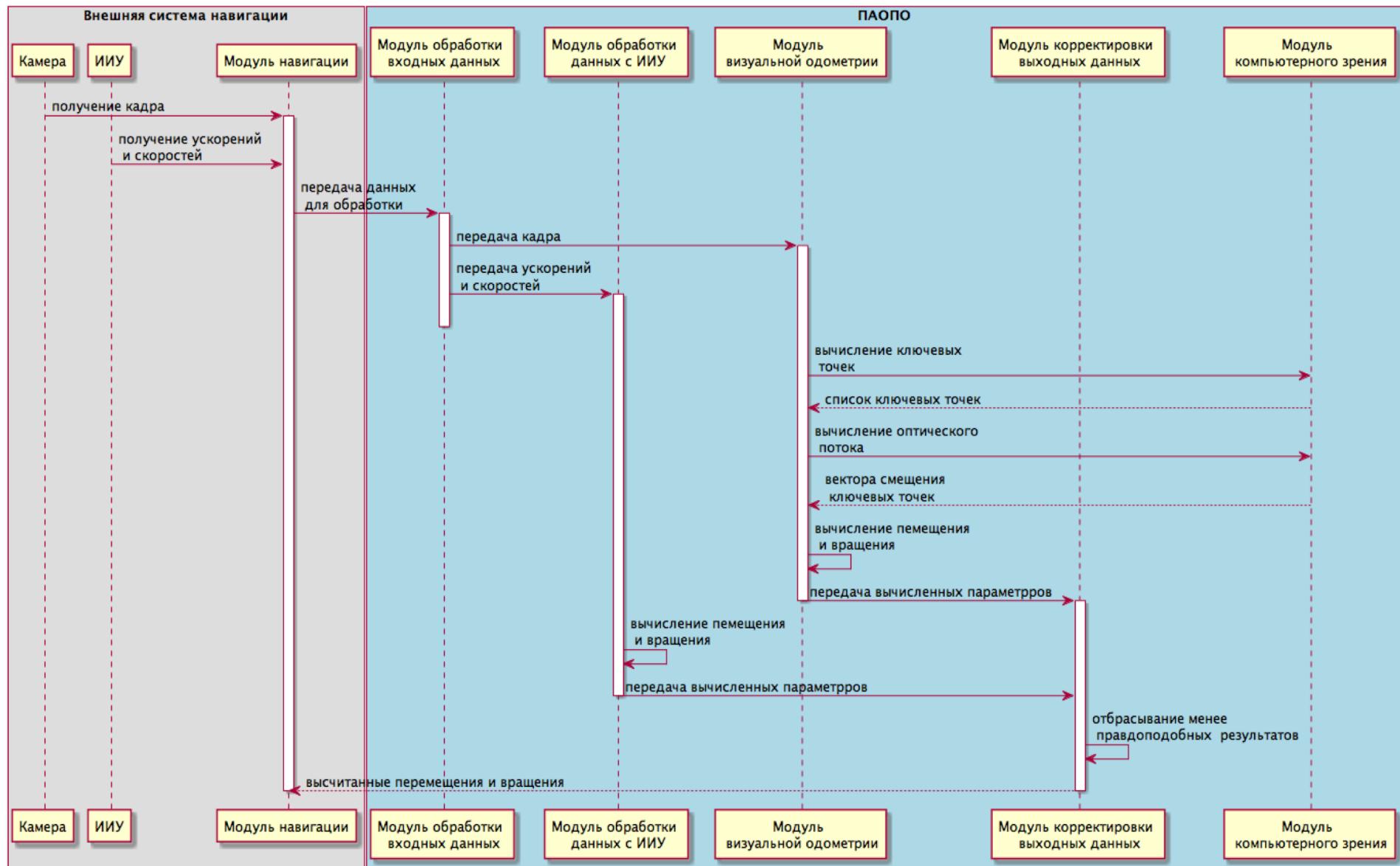
## Задачи, решаемые подсистемой

1. предоставление возможности определять местоположение объекта без связи со спутниками;
2. повышение точности определения местоположения объекта;
3. снижение технических требований к устройствам, осуществляющим выполнение поставленных задач.

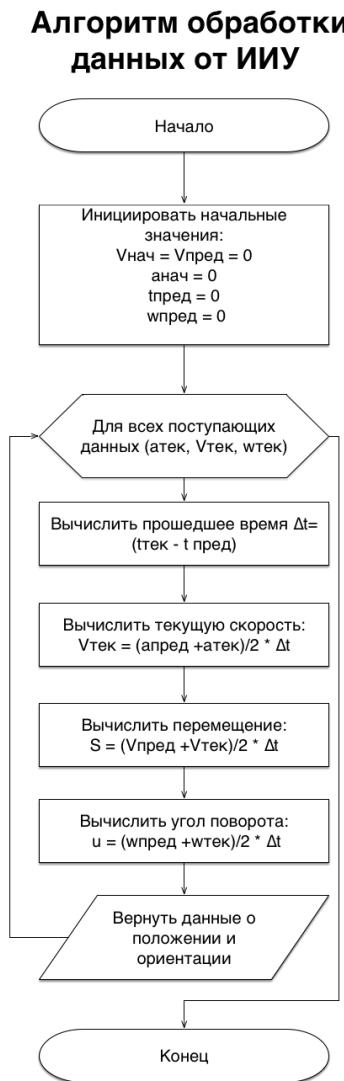
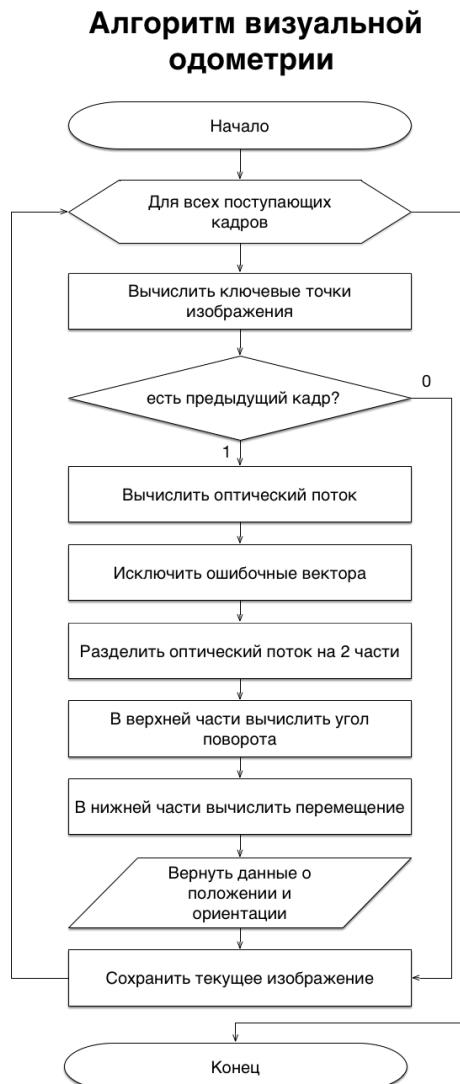
## Сравнительный анализ аналогов и прототипов

Критерий	Вес. коэф.	Проектируемая подсистема	Аналоги		
			ROS odometry	Project tango	SMP Robotics
Необходимость в специальном оборудовании	0,344	1	1	0,4	0,4
Стоимость необходимого оборудования	0,344	1	0,6	0,6	0,1
Сложность интеграции	0,129	1	0,1	1	0,6
Точность	0,129	0,6	1	1	1
Возможность свободного использования	0,054	1	1	1	0,4
Итого:		0,9484	0,7463	0,656	0,4

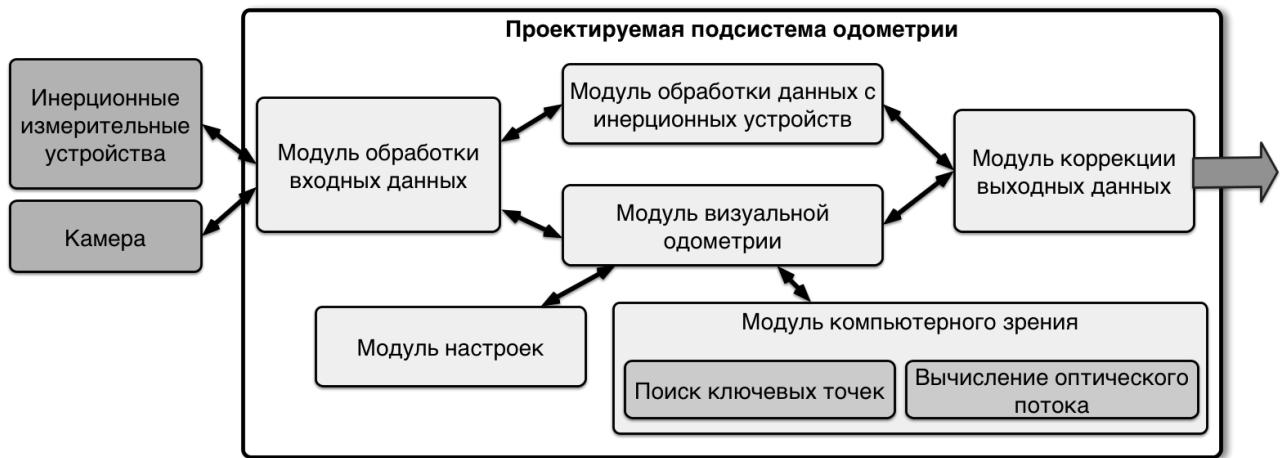
# Диаграмма взаимодействия компонентов



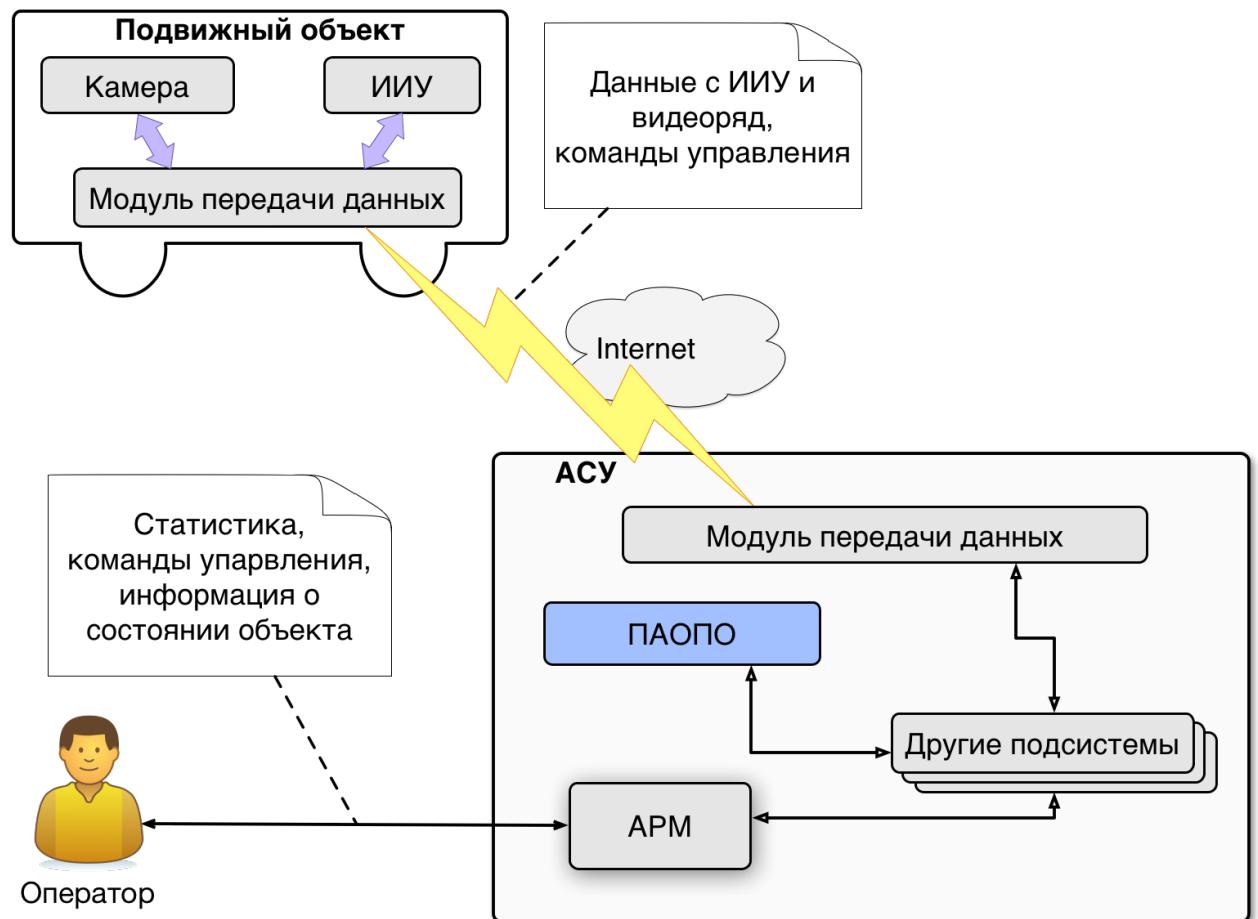
# Блок-схемы алгоритмов



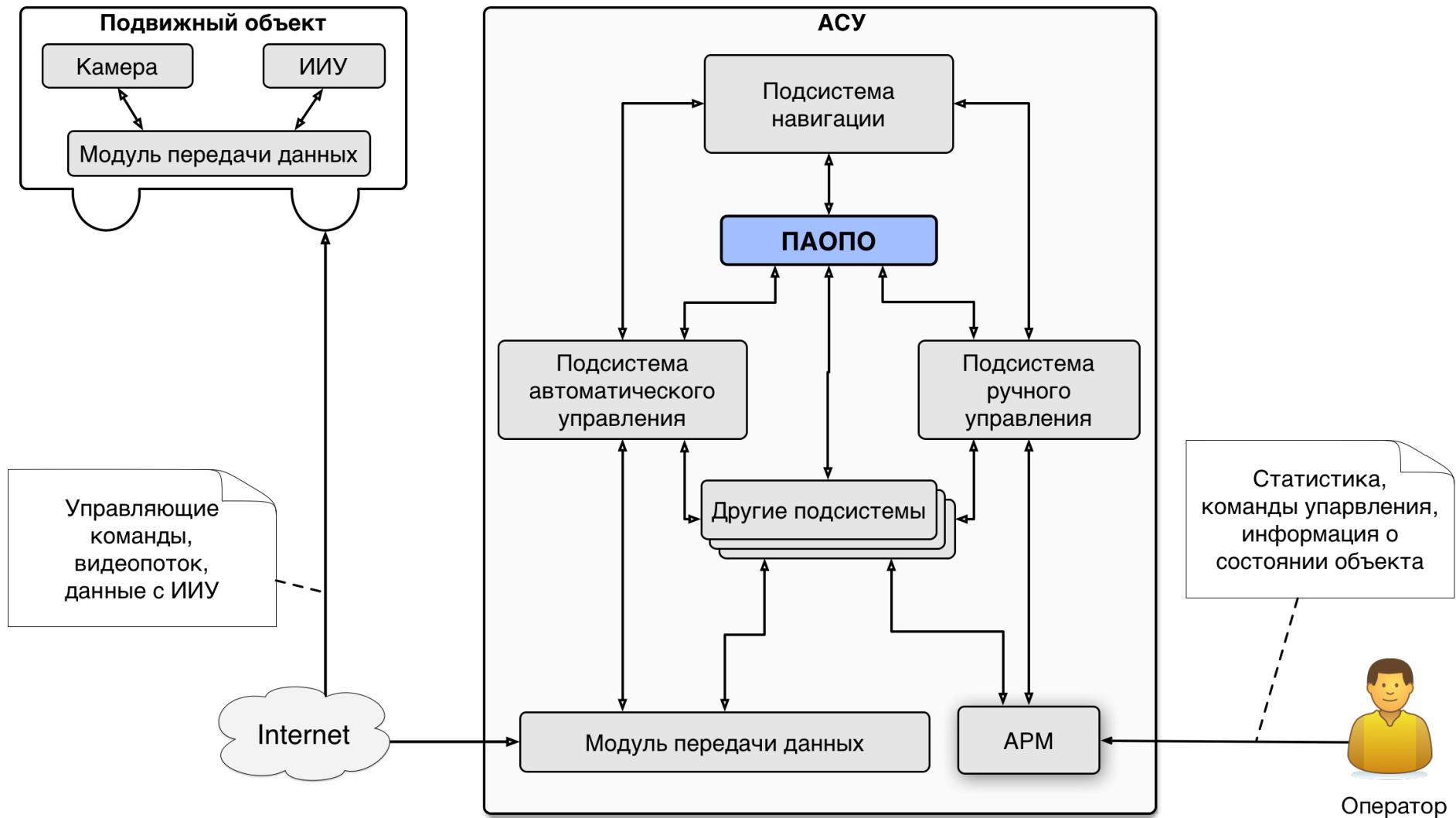
## Структурная схема подсистемы



## ПАОПО в АСУ



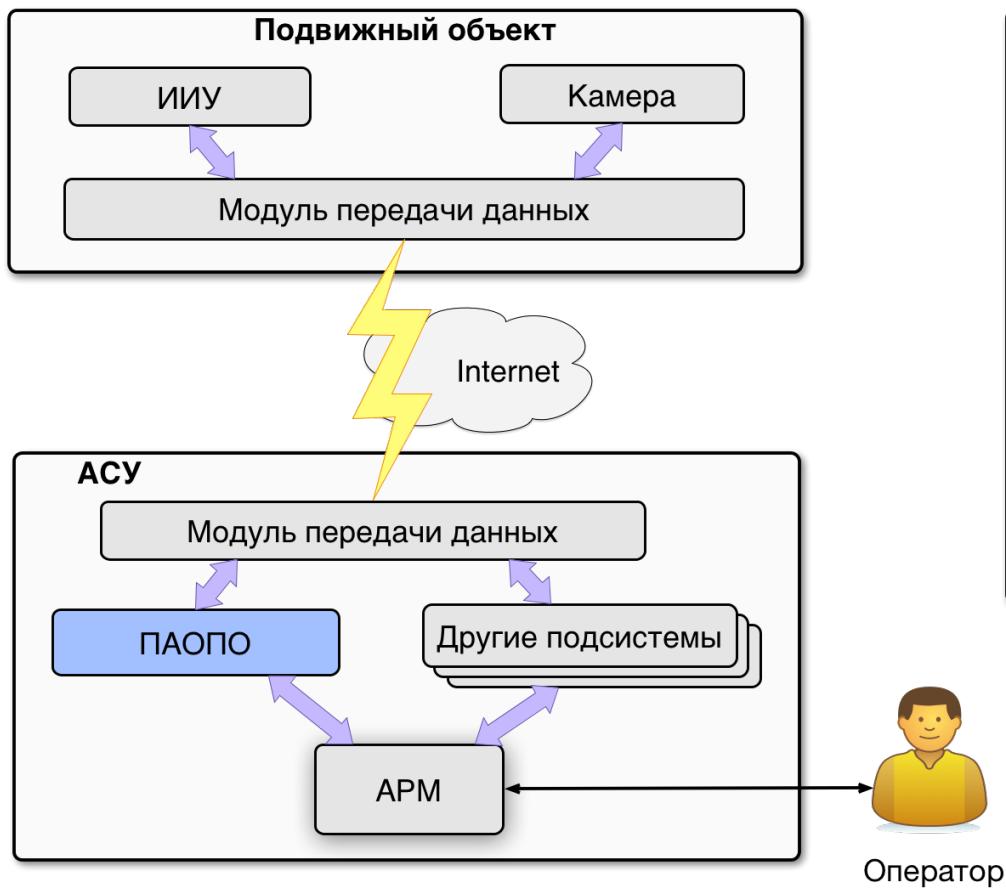
## Подсистема автономного определения перемещения в системах управления подвижными объектами



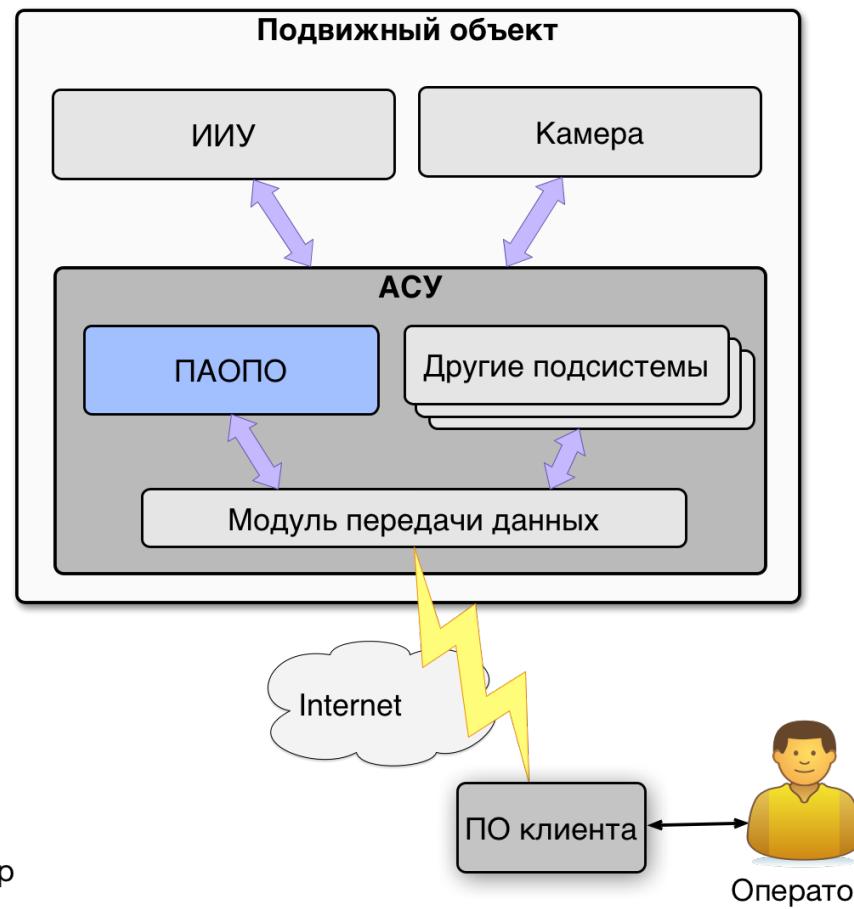
## Схемы возможного размещения подсистемы

Возможно два варианта интеграции системы по ее размещению:

Удаленное размещение



Размещение на платформе



↔ Передача данных по внутренним шинам

⚡ Передача данных с помощью сетевых протоколов (TCP/UDP)

# Выбор весовых коэффициентов критериев сравнения аналогов и прототипов.

Основные этапы метода анализа иерархий.

1) Структуризация процессов принятия решения.

- Выделение вариантов сравнения;
  - Выделение локальных критериев;
  - Формирование глобального критерия;
- 2) Последовательное парное сравнение критериев и вариантов по каждому критерию.
- Для сравнения составляются матрицы сравнения.
- $N$  – матриц для сравнения вариантов по каждому критерию.
- $n$  – количество критериев.
- 3) Вычисление коэффициентов важности сравниваемых величин.
- Определяется собственное значение векторов и их весовые коэффициенты.

Требования к построению матрицы:

- Квадратная и с положительными коэффициентами;
- Симметричные диагональные элементы равны  $K$ ,  $1/K$ ;
- Все диагональные элементы равны 1;
- Для всех элементов должны выполняться условия транзитивности  
(если  $A>B>C$ , то  $A>C$ );
- Условие согласованности;

$$C_i = \sqrt[n]{K_{i1} \cdot K_{i2} \cdot \dots \cdot K_{in}} \quad \alpha_i = \frac{C_i}{\sum_{i=1}^n C_i}$$

4) Проверка согласованности суждений лица принимающих решений. Определяется степень согласованности матрицы сравнений. Если степень согласованности меньше или равна 0,1, то матрица согласованная. Если  $OC > 0,1$ , то матрица не согласованная и следует заново провести повторное сравнение.

$$OC = \frac{(\lambda_{\max} - n)}{(n-1) \cdot R}$$

$R$  – Значение индекса согласованности;

$\lambda_{\max}$  – max соответствия значения матрицы парных сравнений суммы элементов каждого столбца, умножающегося на соответствующий компонент весового фактора из этой же матрицы. Полученные для каждого столбца значения суммируются.  $\lambda_{\max} = \sum_{i=1}^n \lambda_i \cdot \alpha_i$ ;

n	3	4	5	6	7	8	9	10
R	0,58	0,90	1,12	1,24	1,38	1,41	1,45	1,49

Метод анализа иерархии.

Шкала Саати.

Значение	Описание
1	Равны
2	Малое преимущество
3	Среднее преимущество
4	Выше среднего преимущества
5	Сильное преимущество
6	Очень сильное преимущество
7	Значительное преимущество
8	Явное преимущество
9	Абсолютное преимущество

	K1	K2	K3	K4	K5	Ci
K1	1	1	3	3	5	2,141
K2	1	1	3	3	5	2,141
K3	0,33	0,33	1	1	3	0,800
K4	0,33	0,33	1	1	3	0,800
K5	0,2	0,2	0,33	0,33	1	0,337

Сумма: 6,218

$$\lambda_{\max} = 5,034;$$

$$OC = \frac{(5,034 - 5)}{(5 - 1) \cdot 1,12} = \frac{0,034}{4,48} = 0,007589 \leq 0,1$$

$$\alpha_1 = \frac{2,141}{6,218} = 0,344 ;$$

$$\alpha_3 = \frac{0,800}{6,218} = 0,129;$$

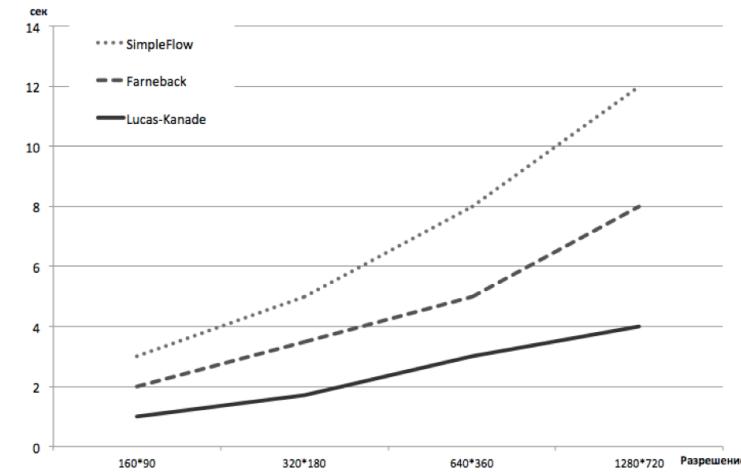
$$\alpha_5 = \frac{0,337}{6,218} = 0,054 ;$$

$$\alpha_2 = \frac{2,141}{6,218} = 0,344 ;$$

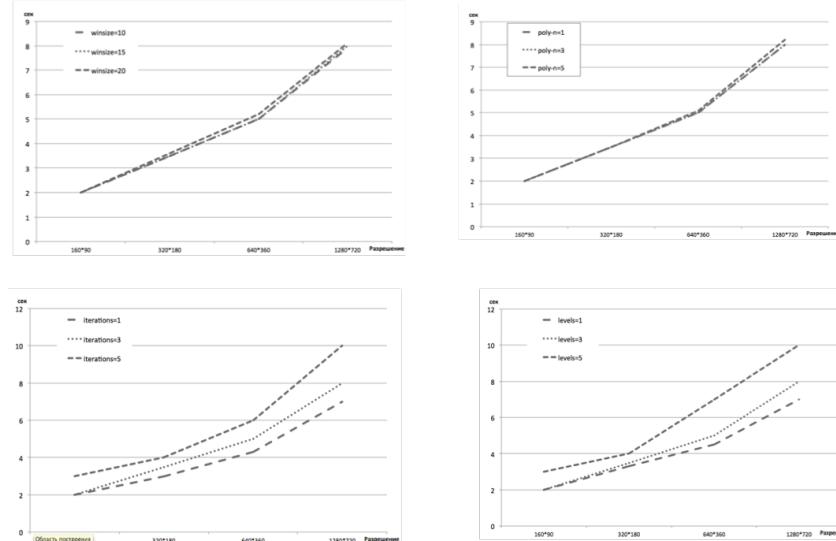
$$\alpha_4 = \frac{0,800}{6,218} = 0,129 ;$$

## Сравнение времени вычисления оптического потока разными методами

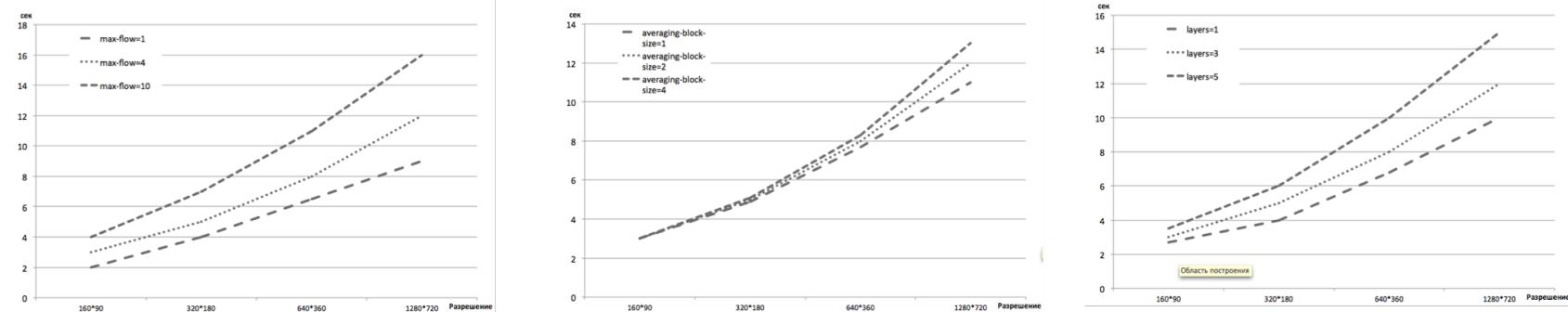
### Сравнение трех основных методов



### Исследование влияния разных параметров на время выполнения FarneBack

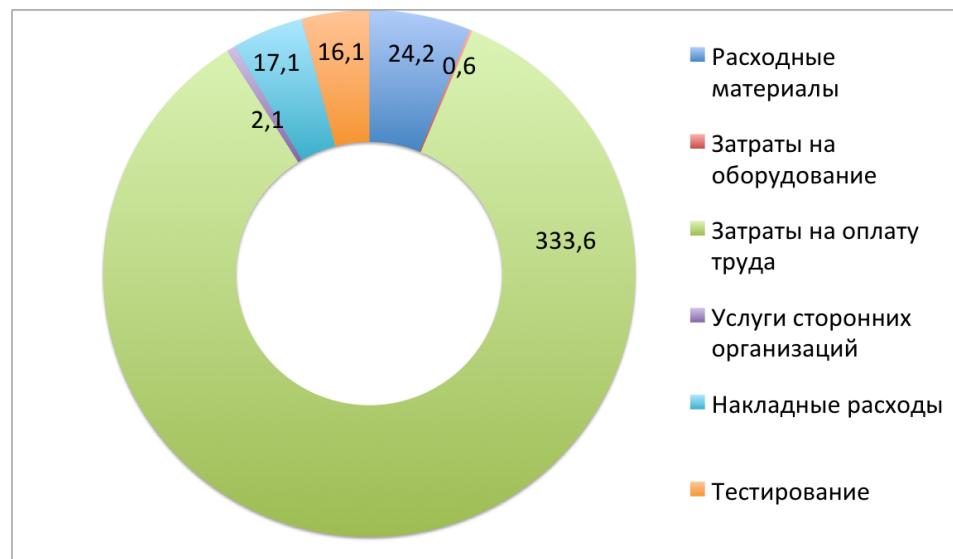


### Исследование влияния разных параметров на время выполнения SimpleFLow



# Организационно-экономическая часть проекта

## Затраты на расходные материалы и инструменты



## Этапы проектирования

1. Формулировка и уточнение задания.
2. Исследование и анализ ПО.
3. Разработка и утверждение ТЗ.
4. Техническое проектирование (разработка моделей данных и алгоритмов).
5. Рабочее проектирование (кодирование и тестирование программных модулей).
6. Тестовые испытания подсистемы.
7. Разработка программной документации.

Выполнил: \_\_\_\_\_ ( Жуков Р.В.)

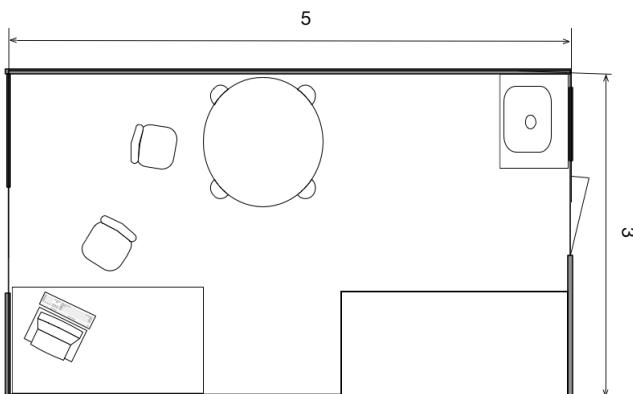
Утвердил: \_\_\_\_\_ (Терехов В.И.)

# Промышленная экология и безопасность

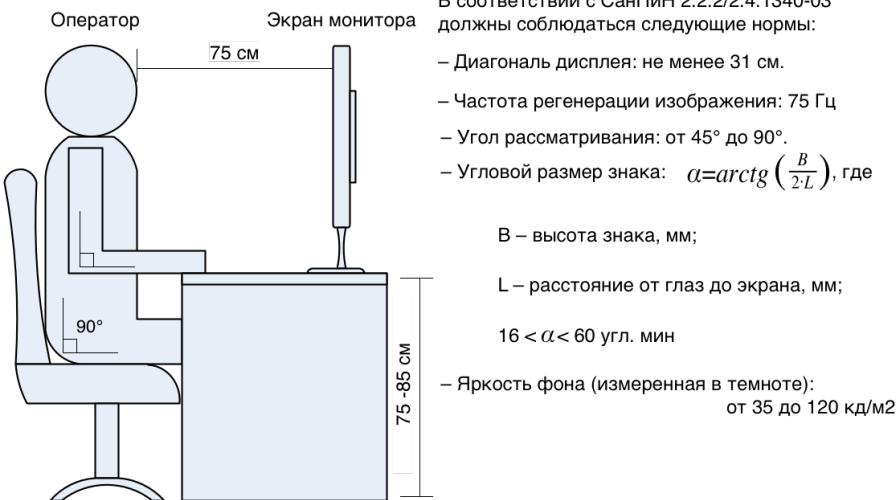
Рабочее место оператора ПЭВМ должно соответствовать требованиям, предъявляемым к нему следующими документами:

- СНиП 23-05-95 «Естественное и искусственное освещение»
- СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»
- СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»

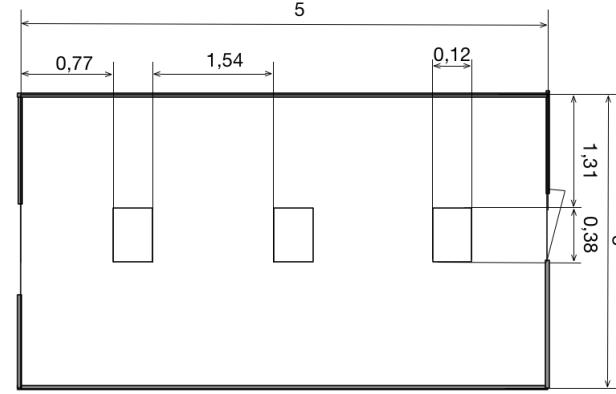
**Схема расположения мебели**



**Организация рабочего места оператора**



**Схема расположения светильников**



**Методика расчета искусственного освещения**

1) Расчет индекса помещения:

$$i = \frac{A \cdot B}{H_c \cdot (A + B)} = \frac{6 \cdot 4}{3,2 \cdot (4 + 6)} = 0,75$$

2) Определяем коэффициент использования светового потока по таблицам в СНиП 23-05-95:  $\eta = 0,4$

$$N = 3 \cdot 2 = 6 \text{ шт.}$$

3) Определяем число светильников:

$$\Phi_{ocb} = \frac{300 \cdot 20 \cdot 1,1 \cdot 1,4}{0,4} = 23\,100 \text{ лк}$$

4) Определяем требуемый общий световой поток:

$$\Phi_{cb} = \frac{\Phi_{ocb}}{N} = \frac{23\,100}{3} = 7700 \text{ лк}$$

5) Определяем световой поток, приходящийся на один светильник:

$$\Phi_a = \frac{\Phi_{cb}}{N} = \frac{7700}{2} = 3\,850 \text{ лк}$$

6) Определяем световой поток, приходящийся на одну лампу:

$$\Phi_a = \frac{\Phi_{cb}}{N} = \frac{7700}{2} = 3\,850 \text{ лк}$$

7) Определим тип лампы исходя из требуемого светового потока:

Тип лампы – люминесцентная ЛБ-65 (световой поток 4550 лм, мощность 65 Вт)