Гироскоп на основе монокулярной камеры

Кудеров П.В.

ИУ9-121

**Оглавление**

[Введение 3](#_Toc420044688)

[1 Аналитическая часть 4](#_Toc420044689)

[1.1 4](#_Toc420044690)

[2 Научно-исследовательская часть 5](#_Toc420044691)

[3 Проектно-конструкторская часть 6](#_Toc420044692)

[4 Технологическая часть 7](#_Toc420044693)

[4.1 Методология разработки и используемые средства 7](#_Toc420044694)

[4.2 Руководство пользователя 7](#_Toc420044695)

[4.2.1 Приложение Gyrocam 7](#_Toc420044696)

[5 Организационно-экономическая часть 10](#_Toc420044697)

[5.1 Введение 10](#_Toc420044698)

[5.2 Организация и планирование процесса разработки программы 10](#_Toc420044699)

[5.2.1 Техническое задание 10](#_Toc420044700)

[Заключение 11](#_Toc420044701)

[Приложение 1. Диаграмма Ганта выполненных работ 13](#_Toc420044702)

# Введение

Последние десятилетия наблюдается высокий уровень интереса к автономной самоуправляемой технике, и с каждым годом он продолжает расти. Робототехническим системам можно найти применение в условиях военных конфликтов, экстремальных для человека или несопоставимых с его габаритами, в качестве транспортных средств для пассажиро- и грузоперевозок, для проведения масштабных и/или простых шаблонных работ и так далее. Одной из наиболее проблемных подсистем самоуправляемой техники является система навигации, и, в частности, подсистема позиционирования.

[*image: nav systems relation*]

Среди популярных применяемых в автономной робототехнике решений – использование инерциальных систем навигации (ИНС). Подобные системы содержат акселерометры для определения параметров линейного ускорения, а также гироскоп (или акселерометры, измеряющие центробежное ускорение) для определения углов поворота и наклона. На основе данных этих датчиков производится последующее вычисление вектора скорости и координат объекта. Преимущества методов инерциальной навигации состоят в автономности, помехозащищенности и возможности полной автоматизации всех процессов навигации. Основная же проблема – наличие дрейфа, то есть накопление ошибки со временем работы. Различными техниками можно уменьшить величину ошибки, но не избавиться вовсе.

Не имеют дрейфа глобальные (спутниковые) системы навигации (GPS, GLONASS, GALILEO и другие их аналоги). Для существующих систем данного класса характерна погрешность позиционирования порядка 5-15 метров (***данные википедии***). Дополнительные проблемы связаны с диапазоном рабочей частоты сигнала – уровень приема сигнала от спутников может серьезно ухудшиться под плотной листвой деревьев, из-за очень большой облачности, в условиях плотной городской застройки. Практически невозможно определить свое точное местонахождение внутри помещений, в тоннеле, причем даже профессиональными геодезическими приемниками (***практически*** ***цитата из ненаучной статьи***). Но и в благоприятных условиях точности в несколько метров для определенного круга задач может быть недостаточно. В таких случаях глобальные системы навигации используются лишь для локализации нахождения объекта, а уточнение координат объекта ведется с помощью других систем.

Среди систем локального позиционирования распространено использование технологий построения трехмерной карты окружающей среды на основе данных эхолокации инфракрасными или ультразвуковыми датчиками.

Набирает популярность использование методов компьютерного зрения. Видеодатчики имеют маленький размер, энергопотребление, цену, поэтому есть интерес использовать их в системах позиционирования и навигации. Для тестирования методов можно использовать даже камеры мобильных телефонов. Системы позиционирования на основе анализа изображений не имеют дрейфа, при этом может достигаться довольно высокая точность, сравнимая с ИНС потребительского класса. Большинство из предлагаемых методов основано на обнаружении базисных элементов изображения и слежения за ними в потоке изображений.

Среди распространенных – метод одновременной навигации и картирования (simultaneous localization and mapping, SLAM) , а также метод определения структуры объекта в процессе движения (structure from motion, SfM). Оба этих метода в той или иной степени пытаются построить 3D модель окружающей среды, относительно которой происходит движение камеры.

[*image: slam & sfm examples*]

\* SLAM – используются Particle filters / Sequential Monte Carlo(SMC) или расширенный фильтр Калмана, bundle adjustment (корректировка узлов?)

\* SfM (Structure from motion) - видимо, то же, что и 3D reconstruction from 2D images, распространен алгоритм Scale-invariant feature transform (SIFT), другой распространенный алгоритм - Speeded Up Robust Features (SURF).

Одна из подзадач, возникающих при построении системы позиционирования с использованием алгоритмов компьютерного зрения – определение углов наклона и поворота камеры, то есть создание аналога гироскопа.

В данной работе реализован метод, предложенный Вилле Хуттуненом и Робертом Пише [**Link1**], который заключается в определения трехмерной ориентации монокулярной камеры с использованием точек схождения перспективы (ТСП), обнаруженных на изображениях.

[*image: mahattan world example, indoor, city outdoor, non-city outdoor*]

Данный метод подходит для использования внутри помещений, в условиях городской застройки, а также мест, где под влиянием деятельности человека четко прослеживаются правильные геометрические примитивы (man-made world, Manhattan world). В таких условиях изобилие линейных объектов правильной геометрической формы позволяет достигнуть высокой точности.

Плюсы конкретно данного метода (***ниже почти дословная цитата из******Link1***):

* в условиях "Manhattan World" изобилие линейных объектов правильной геометрической формы позволяет достигнуть высокой точности
* число интересующих нас ТСП ограничено <= 3
* ТСП не зависят от положения камеры - только от ее ориентации
* робастность относительно случайных нестационарных объектов, попадающих в кадр (люди, транспортные средства и т.п.)

Методы на основе определения ТСП позволяют решать задачи:

* навигации автономных транспортных средств, роботов
* 3D реконструкции окружающей среды
* калибровки камер и коррекции изображений

Метод был протестирован на наборе изображений YorkUrbanDb [**Link3**], созданного на базе York University Centre for Vision Research, а также на случайных изображениях, взятых из интернета или сделанных вручную.

[*image: our result image*]

Набор YorkUrbanDb состоит из 47 изображений внутри помещений и 55 изображений городских сцен с вычисленными значениями ТСП.

Существует целый ряд работ, посвященных вычислению ТСП на снимках городских сцен и внутри помещений, то есть в условиях так называемого «Manhattan World» [**Link2**]..

Промежуточная цель работы состоит в исследовании применимости метода Хуттунена и Пише в условиях съемок внутренних помещений и городских пейзажей, обнаружение слабых мест и ознакомление с аналогичными работами по данной тематике, предложение вариантов улучшения метода, нахождение удобного способа тестирования (?).

Конечная цель – создание библиотеки / приложения, которое позволяет эффективно и с точностью, сравнимой с гироскопами потребительского класса, вычислять углы наклона и поворота камеры по входному потоку изображений в условиях «Manhattan World assumption» в реальном времени..

# Аналитическая часть

## 

# Научно-исследовательская часть

Поставленную задачу можно разделить на следующие этапы:

Используемый в данной работе метод нахождения углов поворота и наклона состоит из следующих шагов:

* выделение сегментов линий на изображении
* выделение наилучших кластеров СЛ, сходящихся в одной точке (или бесконечности), используя алгоритм RANSAC
* (не работает) приведение координат сегментов линий в нормализованные координаты путем домножения слева на обратную матрицу калибровки
* вычисление вектора направления ТСП на основе полученных кластеров СЛ путем приближенного решения задачи минимизации Ax = 0 методом SVD декомпозиции матрицы A
* ортогонализация матрицы, составленной из векторов направлений ТСП
* (нет) получение углов (крен, тангаж, рыскание) из матрицы поворота
* (нет) связывание матриц поворота и кормление их расширенному фильтру Калмана

## Геометрия перспективных изображений

Проективное преобразование и однородные координаты, модель булавочного отверстия, ТСП

## Обнаружение ТСП

**Обнаружение СЛ**

Вкратце преобразование Хо (Хаф, Hough Transform) и анализ связанных компонент ориентаций градиента изображения, возможно про Canny детектор, кратко про LSD (pdf)

Выбрали анализ ... почему?

Lsd - O(img.size()), but slower than Hough

**Кластеризация СЛ**

RANSAC, можно (нужно! но, возможно, не здесь, а дальше в "что можно улучшить") заикнуться про J-Linkage (введение в статье pdf - сравниваются разные способы кластеризации)

RANSAC выбрали, потому что популярное решение.

Расписать функцию расстояния d(vp,l), как в статье, как делаю я.

Расписать как выбирается число итераций алгоритма (вывод формулы)

На данном этапе выработана грубая оценка ТСП

**Уточнение ТСП**

Описать математически текущую ситуацию - пучок линий сходится примерно в одной точке. Какое уравнение связывает их в идеале.

Значит решаем Ax = 0 приближенно. Что делать, если система не переопределена (у меня сейчас просто упадет программа!)?

Какие существуют способы приближенного решения данной задачи? LSQM, SVD, smth else

Почему мы не решаем ее тут, а переходим в нормализованные координаты изображения? Что это такое вообще, зачем? В честь кого это делается? [Cipolla]

Получили нулевые направления, что дальше?

**Определение ориентации по ТСП**

Есть матрица D = [d1, d2, d3]. Если только две приблизительно ортогональны (или 3ей не нашли вообще), 3ю вычисляем на основе их через векторное произведение.

Объясняю, что D скорее всего не ортогональна => можно ортогонализовать. Как? SVD! D' = u \* vt

Приводим матрицу нулевых направлений

Матрица относительного поворота R = D2 D1t

At = A-1, т.к. ортогональная => alpha(R) = alpha(D2) + alpha(D1t) = alpha(D2) - alpha(D1)

# Проектно-конструкторская часть

# Технологическая часть

## Методология разработки и используемые средства

## Руководство пользователя

### Приложение Gyrocam

Результатом работы является Win32 консольное приложение, реализующее алгоритм нахождения точек схождения перспективы (ТСП) на изображении, а также векторов единичных направлений, соответствующих им.

Для использования данного приложения требуется установленная на компьютере пользователя библиотека компьютерного зрения opencv версии 3.0.0. Описание процесса установки и необходимые файлы можно найти по адресу [http://docs.opencv.org/3.0‑alpha/doc/tutorials/introduction/windows\_install/windows\_install.html](http://docs.opencv.org/3.0alpha/doc/tutorials/introduction/windows_install/windows_install.html).

Приложение запускается со следующими аргументами:

1. строка абсолютного или относительго пути к файлу изображения, которое следует обработать. Поддерживается такие форматы как bmp, jpg, jpeg,tiff, png.
2. строка абсолютного или относительного пути выходного файла изображения. Формат выходного изображения будет выбран на основе расширения в имени файла данного пути. Поддерживаемые форматы те же, что и во входном аргументе.

Для изображения, заданного входным аргументом, приложение рассчитывает единичные векторы ТСП и выводит окно. В данном окне отображается само исходное изображение с наложенными поверх него прямыми линиями разных цветов. Пример окна приложения показан на рисунке (**Рисунок 1**).

Сначала поверх изображения накладываются линии, соответствующие трем наилучшим (с вероятностью >= 95%) кластерам сегментов линий, сходящихся в какой-либо одной точке. Для них используются темные варианты трех цветов – темно-синий, темно-зеленый и темно-красный цвета в порядке выделения каждого кластера.

Далее поверх полученного изображения накладываются линии, соответствующие кластерам сегментов линий, сходящихся в вычисленных ТСП. Данные линии отображаются яркими вариантами трех цветов – синим, зеленым и красным (для сравнения, например, если темный вариант красного – bgr(0, 0, 127), то яркий – bgr(0, 0, 255)). Кластеры одного цвета, но разной степени яркости в общем случае могут как полностью совпадать, так и частично различаться. Связь между ними следующая – сегменты темного цвета как бы продуцируют некоторую ТСП, а яркого – индуцированы этой же ТСП. В случае полного совпадения кластеров одного цвета – поверх кластера темного цвета будет наложен кластер яркого цвета и он не будет виден.

Выведенное пользователю изображение также сохраняется на диск в выходной файл. Путь, название и расширение файла определяется соответствующим (вторым) входным аргументом командной строки.



Рисунок 1 – Окно с обработанным изображением, поверх которого наложены сегменты линий, соответствующие найденным ТСП. На нем виден случай, когда кластеры темных вариантов цветов полностью покрыты кластерами ярких цветов.

Помимо описанного выше окна с изображением после вычисления ТСП приложение выводит в консоль некоторую служебную информацию от модуля opencv и затем две матрицы 3х3 с числами с плавающей запятой (**Рисунок 2**). Каждая матрица имеет формат как в примере ниже:

[0.99999738, 0.0022501017, 0.00047868371;

0.93076748, 0.36539984, -0.012442469;

-0.10985497, -0.99394757, -0.00042265668]

Границы матрицы задаются квадратными скобками, строки матрицы разделяются точкой с запятой с последующим переводом строки, числа в рамках одной строки разделяются запятыми.

Первой матрице соответствуют единичные векторы найденных ТСП. Каждый вектор представляет собой строку матрицы. Порядок соответствия векторов ТСП наложенным на изображение линиям следующий – синий, зеленый, красный.

Вторая матрица – ближайшая к первой ортонормированная матрица в смысле нормы Фробениуса. Данная матрица является искомой матрицей поворота системы координат камеры.

Описанный выше вывод двух матриц также сохраняется на диск текстовый файл в том же формате. Полный путь данного файла получается прибавлением расширения «.txt» к строке пути второго входного аргумента (пути к выходному изображению). Например, если при запуске приложения вторым аргументом на вход передать относительный путь «../TestSamples/output.jpg», то матрицы будут записаны в файл по следующему пути: «../TestSamples/output.jpg.txt».



Рисунок 2 – Вывод в консоли матриц направлений ТСП (неортогональной и ортогональной, соответственно) на тестовом запуске.

Описанное консольное приложение было протестировано на наборе изображений YorkUrbanDb (The York Urban Line Segment Database). Набор состоит из 47 изображений внутри помещений и 55 изображений городских сцен. Для каждого изображения из базы приведены данные по точкам схождения перспективы, что позволяет оценить уровень точности работы приложения. Описание и сам набор изображений можно найти на странице базы <http://www.elderlab.yorku.ca/YorkUrbanDB/>.

Также может представлять интерес для тестирования набор изображений в составе так называемой Eurasian Cities Database. Описание и сам архив базы для скачивания доступен на <http://graphics.cs.msu.ru/en/research/projects/msr/geometry>.

# Организационно-экономическая часть

## Введение

Разрабатываемое в рамках дипломной работы программное обеспечение является.

Целью данного раздела является расчет трудоемкости, продолжительности разработки программного обеспечения и сметы затрат.

## Организация и планирование процесса разработки программы

### Техническое задание

.

# Заключение

В данной работе были

Библиография

Link1. Huttunen & Piché, 2012

Link2. Coughlan & Yuille, 1999, 2003

Link3. The York Urban Line Segment Database

1. Диаграмма Ганта выполненных работ