

Санкт-Петербургский государственный университет

Житнухина Мария Андреевна

Выпускная квалификационная работа

Sensors fusion для улучшения выкладывания модели дороги в 3D в ADAS

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование
информационных систем»*

Основная образовательная программа *СВ.5162.2021 «Технологии программирования»*

Научный руководитель:
старший преподаватель кафедры системного программирования, к.т.н., Ю.В. Литвинов

Рецензент:
Доцент СПбПУ, к.ф.-м.н. А.А. Семакова

Санкт-Петербург
2025

Saint Petersburg State University

Maria Zhitnukhina

Bachelor's Thesis

3D roadmodel construction improvement by sensors fusion in ADAS

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5162.2021 "Programming Technologies"*

Scientific supervisor:

C.Sc., senior lecturer Y. V. Litvinov

Reviewer:

C.Sc., docent of SPbPU A. A. Semakova

Saint Petersburg
2025

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор	7
2.1. Способы задания ориентации камеры	7
2.2. Существующие способы определения ориентации камеры	7
2.3. Sensor fusion	9
2.4. Методы синхронизации данных с камеры и IMU	13
2.5. Обзор используемых технологий	14
3. Описание решения	16
3.1. Оценка горизонта по камере	16
3.2. Оценка ориентации по IMU	19
3.3. Оценка сдвига временных меток	20
3.4. Sensor fusion IMU и камеры	21
3.5. Sensor fusion IMU, камеры и колесной одометрии	22
3.6. Применение алгоритма для улучшения перепроецирования	22
4. Эксперимент	24
4.1. Использование алгоритма в симуляторе Webots	24
4.2. Точность алгоритма	24
4.3. Оценка производительности	28
Заключение	29
Список литературы	30

Введение

Согласно исследованиям [11], рынок автомобилей с системами помощи водителю (англ. ADAS, Advanced driver-assistance systems) растёт и прогнозируется рост и в следующие годы.

В таких системах важно уметь правильно проецировать объекты из плоскости изображения в систему координат автомобиля, чтобы точнее строить модель дороги, движения окружающих объектов, отдавать команды при управлении, основываясь на более точной информации об окружающем мире. Чтобы делать проекцию точек на плоскость дороги, необходимо правильно определять ориентацию камеры. При движении автомобиля камера изменяет свое положение относительно дороги, особенно когда транспорт проезжает через «лежачих полицейских» и попадает в ямы, поэтому изменение ориентации необходимо компенсировать для более точного проецирования.

Положение камеры можно задать с помощью трех углов Эйлера¹: тангажа (англ. pitch), рыскания (англ. yaw) и крена (англ. roll). Тангаж меняется при тряске, рыскание при повороте машины, а крен стабилен, если машина не начинает переворачиваться и не проезжает ямы. Для уточнения тангажа и рыскания используются разные наборы сенсоров из-за различий в физических силах, влияющих на изменения этих углов.

Если известен уровень горизонта, то можно определить наклон камеры к дороге. Компенсация тряски — определение изменения уровня горизонта, то есть положения прямой на изображении, на которой расположены все точки схода пучков параллельных прямых, лежащих в плоскости дороги. Компенсацию тряски можно делать различными способами и с помощью различных датчиков. Наиболее распространенные — при помощи камеры или при помощи инерциального измерительного модуля (англ. IMU, Inertial measurement unit).

Стабилизация изображения с помощью камеры чаще всего осуществ-

¹Sciencedirect, URL: <https://www.sciencedirect.com/topics/engineering/euler-angle> (дата обращения: 18.11.2024)

ляется при помощи измерения оптического потока², но одних данных с камеры не всегда достаточно. Например, алгоритм не работает, если нет удаленных объектов на уровне горизонта — такое может произойти, если перед автомобилем тупик. Помимо этого, алгоритм будет плохо работать в темноте.

При использовании данных с IMU для оценки ориентации автомобиля, едущего по дороге, возникает проблема зашумленности данных, особенно значения угла тангажа, присутствуют погрешности в проставлении временных меток.

Для оценки рыскания в том числе используют колесную одометрию, то есть используется информация о разности пройденного угла колесами. Точность этого метода ограничена энкодером колеса, и, как и в случае с IMU, измерения не происходят одновременно с отправлением изображения.

Данные, которые можно получить от каждого из имеющихся на машине сенсоров, зашумлены. Можно добиться более точной оценки ориентации камеры, если использовать несколько источников информации — такой подход называется *sensor fusion*³. Он сопряжен с рядом сложностей: информация может поступать не одновременно, временные метки для пришедших данных проставляются с погрешностью. Уже существуют алгоритмы, объединяющие данные с разных датчиков, но в открытом доступе не описан алгоритм, который одновременно использует камеру, колесную одометрию и инерциальный измерительный модуль для *sensor fusion*. Реализация такого алгоритма для трех датчиков является целью данной работы.

²Документация OpenCV, URL: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html (дата обращения: 18.11.2024)

³Устоявшегося термина в русскоязычной среде на момент 19.11.2024 нет

1. Постановка задачи

Целью работы является реализация алгоритма sensor fusion данных с IMU, колесной одометрии и камеры для определения ориентации камеры автомобиля. Для её выполнения были поставлены следующие задачи.

- Произвести обзор существующих решений.
- Реализовать алгоритм определения ориентации по IMU.
- Реализовать алгоритм определения ориентации по данным с камеры.
- Реализовать алгоритм sensor fusion.
- Протестировать алгоритм.

2. Обзор

В системах помощи водителю важна точность проецирования из плоскости изображения на плоскость дороги. Она зависит от ряда параметров, и один из них — определение ориентации камеры. Угол между камерой и дорогой меняется во время езды из-за тряски, вызванной неровностями поверхности, по которой происходит движение. К примеру, если не вычислять ориентацию камеры при проезде через лежащих полицейских, параллельные линии разметки при проецировании в вид сверху не будут параллельными.

Далее будут описаны различные способы определения изменения ориентации камеры при движении автомобиля при помощи одного или нескольких датчиков.

2.1. Способы задания ориентации камеры

Существуют два основных способа задать ориентацию объекта: при помощи углов Эйлера⁴ и кватернионов⁵.

Углы Эйлера — три угла, описывающие поворот тела вокруг трех осей, как показано на рис. 1. Для углов Эйлера важен порядок, в котором совершаются повороты вокруг осей.

Кватернион представляет собой четырехмерный вектор, однозначно задающий ориентацию тела в пространстве.

2.2. Существующие способы определения ориентации камеры

Для определения ориентации может использоваться камера, IMU, а также угол поворота может быть оценен по колесной одометрии. Каждый из датчиков оценивает ориентацию либо частично, либо не во всех

⁴Sciencedirect, URL: <https://www.sciencedirect.com/topics/engineering/euler-angle> (дата обращения: 18.11.2024)

⁵Статья в Wikipedia, URL: <https://en.wikipedia.org/wiki/Quaternion> (дата обращения: 18.11.2024)

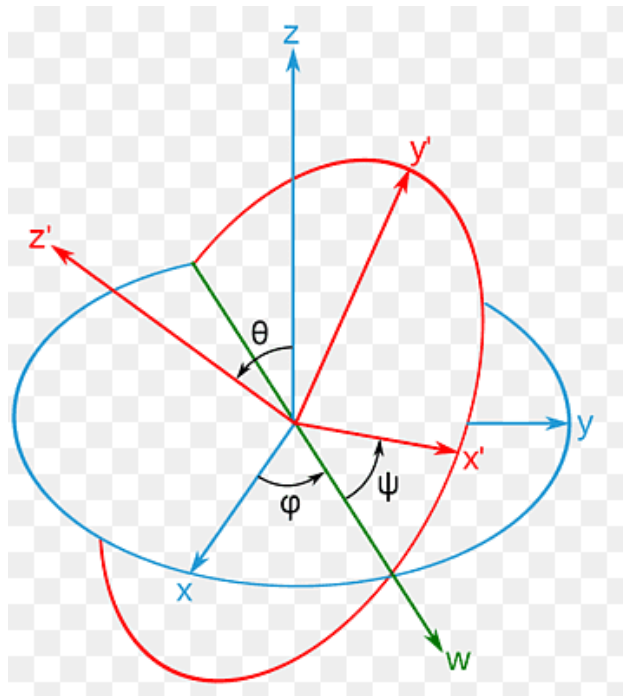


Рис. 1: Углы Эйлера

ситуациях, либо данные могут задержаться относительно камеры. Можно применить *sensor fusion* — подход, при котором одновременно используются показания нескольких датчиков для получения более точного результата.

2.2.1. Определение ориентации при помощи данных с камеры

Существуют различные способы оценки ориентации при помощи видеопотока. Камеры делятся на два типа: стереокамеры и монокамеры. Стереокамера состоит из двух камер и работает по схожему принципу, что и пара человеческих глаз. Она позволяет оценивать расстояние до точек в пространстве, однако стоит намного дороже обычной камеры, и при больших скоростях теряется точность вычисления расстояний из-за рассинхронизации двух изображений.

Основным способом оценки изменения ориентации камеры по видео является использование оптического потока (англ. *optical flow*)⁶. Данный алгоритм позволяет сопоставить точкам с предыдущего кадра точ-

⁶Документация OpenCV, URL: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html (дата обращения: 18.11.2024)

ки на следующем. Алгоритм работает при выполнении следующих условий.

1. Вектора перемещений соседних точек близки.
2. Яркости пикселей не меняются.

Общим недостатком для всех камер при оценке ориентации является то, что они плохо работают в условиях плохой освещенности, при плохой видимости из-за осадков. Так же сложно оценить измерение углов, когда в поле видимости нет далеких объектов, например, если автомобиль находится в пробке, или впереди него большое здание. Частота обновления ориентации ограничена частотой камеры.

2.2.2. Использование данных с IMU

IMU (Inertial Measurement Unit) — инерционный измерительный блок. Данный сенсор является комбинацией нескольких других, как правило, акселерометра, гироскопа и магнитометра. С его помощью можно оценить ориентацию датчика в мировой системе координат. Его частота намного превышает частоту камеры.

2.2.3. Использование колесной одометрии

Одометрия — использование данных о движении приводов для оценки перемещения. В случае колесной одометрии, данные берутся с колесных приводов. По разности значений на энкодерах на колесах можно узнать, на какой угол повернула машина. Недостатком этого метода является то, что он не учитывает проскальзывание колеса, которое может происходить на мокрой или оледеневшей поверхности, а также то, что данный метод зависит от диаметра колес и шин на них.

2.3. Sensor fusion

Sensor fusion — процесс объединения данных от нескольких датчиков для получения более точного результата. Существуют различные подходы: с помощью фильтра Калмана, с помощью нейронных сетей [14],

с помощью придавания разных весов датчикам. Существуют и другие методы: например, с помощью Байесовского подхода [16], с помощью графов [7], с помощью фильтров [6], с помощью нечеткой логики (англ. fuzzy logic) [10], но они применяются, как правило, для другой задачи — определения положения в пространстве.

Фильтр Калмана [3] позволяет, зная закон, по которому изменяется величина, а также показания измерительного прибора, получить приближение этой величины. Помимо этого, предполагаем, что ошибки сенсора и модели — случайные величины с математическим ожиданием, равным нулю, а их дисперсии известны. Показаниям сенсора придается вес, который итерационно меняется по ходу вычислений. Данный фильтр применяется для линейных систем, тогда как для более сложных применяется расширенный фильтр Калмана [12] (англ. EKF, Extended Kalman Filter). Его основное отличие от фильтра Калмана заключается в том, что изменение состояния системы описывается нелинейным уравнением.

2.3.1. «Sensor data fusion using Kalman filter»

В статье «Sensor data fusion using Kalman filter [13]» описываются четыре типа sensor fusion колесной одометрии и ультразвукового датчика для определения положения автомобиля.

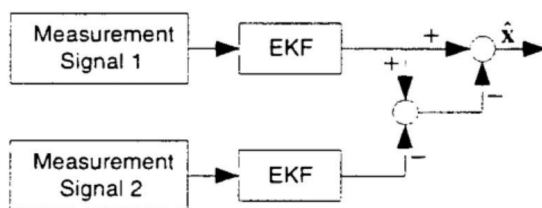


Рис. 2: Сравнение результатов после EKF

- Оба сигнала фильтруются при помощи EKF. Затем результаты сравниваются, и ошибка используется для коррекции измеряемого сигнала, как показано на рис. 2. $\hat{x}$ — выход алгоритма, в уз-

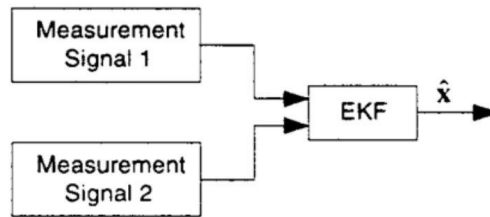


Рис. 3: Данные с двух датчиков используются в EKF

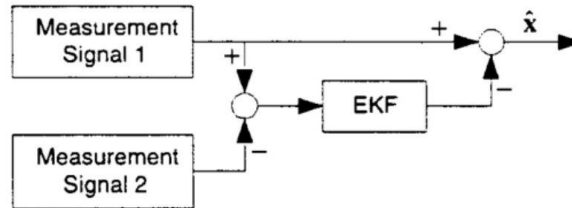


Рис. 4: Выбирается, какие данные фильтровать, обратное распространение ошибки

лах обозначенных кругами происходит обработка двух значений с предыдущих шагов алгоритма.

- Данные комбинируются перед фильтрацией с помощью EKF, как показано на рис. 3.
- Сигналы сравниваются, и одно из значений фильтруется с помощью EKF. После этого ошибка учитывается в одном из измеренных сигналов, как показано на рис. 4.
- Сигналы сравниваются, и одно из значений фильтруется с помощью EKF. После этого ошибка отправляется обратно в один из сигналов и в качестве итогового результата отображается исправленное значение, как показано на рис. 5.

Однако в данной работе не решается задача определения ориентации, а также объединяются данные только с двух датчиков.

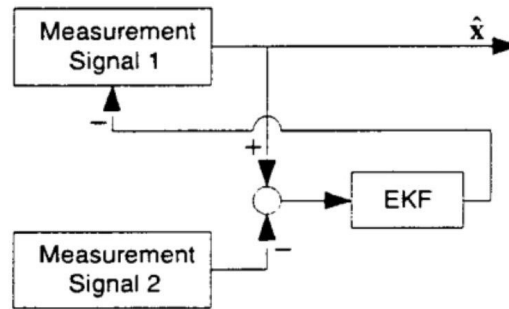


Рис. 5: Выбирается, какие данные фильтровать, после обратного распространения ошибки корректируется отравляемое значение

2.3.2. «Improving Orientation Estimation in Mobiles with built-in Camera»

Авторы «Improving Orientation Estimation in Mobiles with built-in Camera [4]» предлагают экстраполировать данные, получаемые при помощи алгоритма optical flow в соответствии с частотой IMU. С помощью фильтра Маджвика получаем ориентацию по гироскопу и акселерометру. Данные IMU пропускаются через фильтр высоких частиц, а результаты, полученные при помощи камеры, пропускаются через фильтр нижних частиц. Фильтр высоких частиц фильтрует данные по порогу сверху, тогда как фильтр нижних частиц по порогу снизу.

Для описания ориентации используются кватернионы. Для sensor fusion используется линейная формула, использующая результаты обоих датчиков, которые прошли фильтрацию.

2.3.3. «Data Fusion Of Stereo Vision and Gyroscope For Estimation Of Indoor Mobile Robot Orientation»

В статье «Data Fusion Of Stereo Vision and Gyroscope For Estimation Of Indoor Mobile Robot Orientation [8]» предлагается использовать стереозрение и гироскоп для оценки ориентации робота. Делается sensor fusion при помощи фильтра Калмана. Поскольку частота гироскопа больше, чем частота камеры, то пока нового кадра ещё нет, ориентация по кадрам определяется только по данным с гироскопа.

2.3.4. Прочие статьи

В статье «Visual Gyroscope: Integration of Visual Information with Gyroscope for Attitude Measurement of Mobile Platform [15]» используется фильтр Калмана, в котором матрица процесса меняется в зависимости от того, доступны ли результаты вычисления ориентации по камере.

В статье «Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation [5]» применяется расширенный фильтр Калмана для sensor fusion данных с камеры и IMU.

2.4. Методы синхронизации данных с камеры и IMU

Поскольку результаты измерений не могут быть получены одновременно и временные метки могут проставляться разными способами, важным аспектом в sensor fusion является синхронизация данных с разных датчиков.

Всем данным сопоставляются временные метки. Они могут высчитываться на основе частоты датчика, либо отдельно для каждого сообщения.

В статье о синхронизации данных [1] автор определяет временные метки для кадров, вычисляя частоту камеры и сдвиг относительно IMU при помощи фильтра частиц. Когда доступны новые данные IMU, проверяется, нет ли новых данных о движении камеры. Когда так только оба списка данных с датчиков содержат новую информацию, они интерполируются, нормализуются и вычисляется корреляция между двумя движениями.

Оценка кросс-корреляции двух временных рядов, сдвигая один из них, позволяет оценить сдвиг во времени между ними, таким образом появится возможность сопоставить последнему входящему изображению с временной меткой данные от IMU с временной меткой. Чем бли-

же значение корреляции к единице, тем более похожи ряды.

Для оценки частоты кадров в упомянутой выше статье используется фильтр частиц. При фильтре частиц формируется несколько гипотез оцениваемого параметра, и из них выбирается наиболее правдоподобная. Правдоподобность частоты кадров оценивается по тому, сколько кадров удовлетворяли ей по истечении фиксированного времени.

Помимо синхронизации данных с помощью кросс-корреляции, существует физическая синхронизация данных, но она не везде поддерживается на аппаратном уровне.

2.5. Обзор используемых технологий

При реализации был использован язык C++ 17 стандарта, библиотека OpenCV версии 4.10.0 для работы с изображениями и библиотека Eigen для работы с кватернионами. Далее описывается, почему были выбраны перечисленные технологии.

2.5.1. C++

Для реализации используется язык C++, поскольку на нём можно писать достаточно производительный код для использования в системах реального времени, и на нём принято вести разработку в данной предметной области.

2.5.2. OpenCV

Библиотека OpenCV является де-факто стандартом для прототипирования и разработки алгоритмов в области компьютерного зрения, не имеет прямых аналогов, и по этой причине используется для решения поставленной задачи.

2.5.3. Eigen

Для работы с кватернионами на C++ была использована библиотека Eigen. Среди возможных альтернатив можно выделить OpenGL. Они

обе удовлетворяют требованию возможности работы с кватернионами, но OpenGL более тяжеловесная и больше предназначена для работы с графикой.

2.5.4. Способ передачи данных

Для коммуникации с устройствами, отправляющими данные с датчиков, используется CycloneDDS⁷. DDS (англ. Data Distribution Service, Служба распространения данных) инструмент для распределенных систем, позволяющий обмениваться сообщениями по модели публикация-подписка.

⁷Официальный сайт CycloneDDS, URL: <https://cyclonedds.io/>, (дата обращения: 07.05.2025)

3. Описание решения

На данный момент было реализовано вычисление уровня горизонта по данным с камеры и IMU. Множество всех точек схода всех пучков параллельных прямых в плоскости дороги образует прямую в плоскости изображения, и эта прямая является линией горизонта. Важно отметить, что по изменению горизонта можно узнать изменение тангажа и наоборот.

3.1. Оценка горизонта по камере

В данном решении алгоритм оценки горизонта основан на вычислении оптического потока (англ. optical flow) и кластеризации с помощью плотностного алгоритма DBSCAN.

Изображения переводятся в тона серого, чтобы по ним можно было вычислить оптический поток методами, встроенными в OpenCV. Чтобы использовать оптический поток, необходимо выбрать особые точки, которые будут отслеживаться между двумя кадрами.

Особые точки выбираются как объединение двух множеств особых точек, чтобы они не были сконцентрированы в одной части кадра. Все точки выбираем в прямоугольнике высотой 100 пикселей с центром на последней линии горизонта и шириной как у исходного изображения.

1. Часть точек выбирается с помощью `cv::goodFeaturesToTrack`, метода, встроенного в библиотеку OpenCV.
2. Вторая часть точек состоит из локальных максимумов градиента на четырех прямых, параллельных линии горизонта, и берется по одной точке на промежутке в 50 пикселей. Градиент вычисляется при помощи оператора Собеля. Оператор Собеля — дискретный дифференциальный оператор, вычисляющий приближение градиента яркости изображения.

Чтобы оценивать сдвиг относительно бесконечно удалённого горизонта более точно, среди найденных точек необходимо выбрать те, что

лежат на удалённых неподвижных объектах: не на пешеходах, не на машинах, не на деревьях вдоль дороги. Таких объектов в кадре много. Чтобы минимизировать отслеживание точек на них, было решено использовать алгоритм кластеризации для поиска кластера с наиболее подходящими точками. Выбирается наибольший кластер: отдельно движущиеся небольшие объекты выделяются в отдельные кластеры.

Был выбран алгоритм кластеризации DBSCAN (англ. Density-based spatial clustering of applications with noise, что можно перевести как «основанная на плотности пространственная кластеризация для приложений с шумами») по той причине, что он позволяет не указывать количество кластеров, так как оно заранее неизвестно. Помимо этого, он распознает выбросы, его вычислительная сложность составляет $O(n \cdot \log n)$, он работает достаточно быстро на системе реального времени.



Рис. 6: Пример работы кластеризации

Алгоритм DBSCAN получает на вход множество точек, и каждой из них либо присваивается номер кластера, либо присваивается значение, соответствующее выбросу.

Параметрами алгоритма являются целое число N и вещественное число ϵ . В начале алгоритма ищем точку, в ϵ окрестности которой находится хотя бы N других точек, их помечаем как посещённые и сохраняем в одном кластере. Эта точка называется корневой, а её соседи достижимыми. Среди достижимых вершин в кластере ищем корневые точки, и для каждой из них повторяем описанный процесс, добавляя



Рис. 7: Пример работы кластеризации

достижимые точки в кластер. Поскольку точек конечное число, то в какой-то момент мы больше не сможем добавлять точки в данный кластер, и начнём искать новый кластер среди не посещённых точек. В качестве метрики используется максимум модулей координат x и y разности точек.

В данном случае на вход подаются вектора сдвигов, вычисленные в результате работы оптического потока, а также экспериментально подобранные значения $N = 8$, $\epsilon = 0.2$ при размере изображения 920×620 . При увеличении сторон изображения в k раз ϵ увеличиваем в k раз. На рис. 6, рис. 7 можно увидеть, как разными цветами отмечены точки, соответствующие разным кластерам. Черные точки являются выбросами.

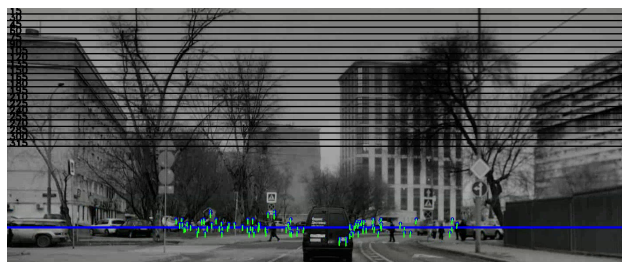


Рис. 8: Пример выделенного кластера

Среди кластеров выбираем кластер максимального размера, и среди точек кластера находим точку с медианным значением по вертикальной оси y . Этот сдвиг добавляем к накопленному сдвигу. Накопленный

сдвиг — сумма всех сдвигов горизонта между соседними кадрами за время работы алгоритма. Пример выделенного кластера с увеличенными в 10 раз сдвигами можно увидеть на рис. 8.

Если в качестве результата возвращать накопленный сдвиг, складывая посчитанные сдвиги между соседними изображениями, то ошибка вычисления будет увеличиваться, и значение горизонта быстро станет неправдоподобным.

Обозначим накопленный сдвиг за *accum_shift* и будем хранить историю последних сдвигов с помощью типа данных *circular_buffer*⁸ из библиотеки *Boost*. Было решено хранить 70 последних значений *accum_shift*, обозначим *shifts*, 40 последних средних значений *accum_shift*, обозначим *means*. Из текущего значения *accum_shift* вычитается среднее арифметическое из четырех минимальных по модулю элементов *means*, к нему прибавляется изначальное значение горизонта, и это число является выходом алгоритма.

3.2. Оценка ориентации по IMU

От инерциально-измерительного модуля приходят показания акселерометра и гироскопа, временная метка сообщения, а также временная метка, вычисленная по частоте датчика и счетчику, которая является более точной, чем время отправки сообщения.

При старте алгоритма вычисляется вектор среднего отклонения от нуля гироскопа по трём осям в состоянии покоя. Для вычисления используются 200 показаний датчика. В дальнейшем сдвиг вычитается из новых полученных данных.

Чтобы получить по входным данным ориентацию IMU как кватернион, был использован фильтр Мэджвика, использующий показания гироскопа и акселерометра.

Фильтр Мэджвика корректирует показания гироскопа при помощи акселерометра. Определить ориентацию по акселерометру возможно

⁸Документация Boost, URL: https://www.boost.org/doc/libs/1_61_0/doc/html/circular_buffer.html (дата обращения: 23.12.2024)

благодаря распределению гравитации по его осям. Это возможно, когда внешние ускорения, действующие на датчик, пренебрежимо малы по сравнению с гравитационным.

Поскольку фильтр Мэдживика не ожидает внешнего ускорения от акселерометра, его приходится минимизировать. Для этого после каждой итерации фильтрации вычисляется соответствующий вектор показаний акселерометра, который могла бы дать гравитация. Из этого вектора вычитается вектор показаний акселерометра. Таким образом, мы получаем вектор внешнего ускорения. На следующей итерации фильтрации он вычитается из показаний акселерометра.

3.3. Оценка сдвига временных меток

Раз в 2000 кадров вычисляется сдвиг временных меток IMU относительно временных меток камеры.

Сравнивается временной ряд производной угла тангажа, полученного по данным с камеры, и временной ряд изменений угла с гироскопа.

Угол тангажа вычисляется по уровню горизонта на изображении при помощи внутренних параметров камеры по следующей формуле:

$$pitch = atan((hor - pp_y)/f_y),$$

где $pitch$ — угол тангажа, hor — уровень горизонта на текущем кадре в пикселях, pp_y — оптический центр камеры в пикселях, а f_y — фокальное расстояние камеры.

Вычисляется дискретная производная этого угла по двум соседним значениям.

Перебирается сдвиг от -0.5 секунд до +0.5 секунд и прибавляется к временным меткам IMU с шагом 0.02 секунды.

Частота IMU и камеры отличается, поэтому производится линейная интерполяция угла с IMU по временным меткам с камеры, после чего к обоим рядам применяется одномерный фильтр Гаусса. После применения фильтра вычисляется значение функции корреляции полученных рядов, и выбирается значение сдвига, при котором корреляция ближе



Рис. 9: График зависимости кросс-корреляции от сдвига временных меток

всего к единице, пример можно увидеть на рис. 9.

Вычисление сдвига между IMU и колесной одометрией делается аналогично. Основное отличие заключается в том, что интегрируются показания гироскопа и используется угол рыскания. Вычислять производную угла колесной одометрии затруднительно, так как изменения показаний колесной одометрии происходят скачками из-за шага энкодеров на колесах.

3.4. Sensor fusion IMU и камеры

Вычисляется сдвиг между временными метками камеры и IMU, прибавляется к временным меткам IMU.

Для объединения данных с IMU и камеры используется вариация

фильтра Мэджвика, в которой используются акселерометр, гироскоп и магнитометр. Когда появляется новый кадр, производится итерация фильтрации, где вместо магнитометра используется угол, полученный с камеры, а также два угла с предыдущего измерения. В промежутках между кадрами угол обновляется без использования камеры и магнитометра.

3.5. Sensor fusion IMU, камеры и колесной одометрии

Вычисляется сдвиг временных меток IMU и колесной одометрии, уже вычислен сдвиг между IMU и камерой. Эти два сдвига прибавляются к временным меткам колесной одометрии, и все датчики оказываются синхронизированы с камерой.

Используется комплементарный фильтр для объединения результата sensor fusion камеры и IMU с колесной одометрией.

3.6. Применение алгоритма для улучшения проецирования

Реализованный алгоритм был использован для улучшения качества проецирования в нескольких приложениях.

- Повышение точности проецирования камерных детекций машин в мировую систему координат.
- Улучшение проецирования четырех камер кругового обзора в вид сверху. Изображения с четырех камер трансформируются в изображение, которая могла бы видеть виртуальная камера, находящаяся над машиной.
- Улучшение проецирования камерных детекций полос в систему координат автомобиля.

Улучшение качества проецирования было оценено визуально по причине отсутствия подходящих эталонных данных.

4. Эксперимент

4.1. Использование алгоритма в симуляторе Webots

Webots⁹ — 3D-симулятор для робототехники с открытым исходным кодом. Он был выбран, поскольку способен генерировать изображение с камеры и позволяет регулировать ее внутренние параметры, а также способен генерировать данные гироскопа, акселерометра и колесной одометрии. Для интеграции с ним была написана прослойка, называемая контроллером¹⁰. Для её реализации использован язык C++. В контроллере принимаются данные колесной одометрии, изображение с камеры, внутренние параметры камеры (размер изображения, фокальное расстояние, оптический центр), показания IMU от симулятора и отправляются при помощи CycloneDDS для дальнейшего использования реализованным алгоритмом.

Также был создан ландшафт, напоминающий городской, на основе шаблона «City»¹¹, встроенного в Webots.

4.2. Точность алгоритма

По той причине, что нет эталонных данных ориентации автомобиля, эксперимент был проведён в симуляторе. Эталонные данные не могут быть получены из-за шумов, возникающих на акселерометре и гироскопе. В качестве платформы для симулятора был использован Webots.

Была выбрана данная платформа по той причине, что в ней имеются библиотеки для генерации данных с IMU, колесной одометрии и камеры, поддерживается работа с автомобилями.

В качестве эксперимента алгоритм sensor fusion был запущен внутри симулятора Webots, см. рис 10, 12.

Средняя ошибка определения угла тангажа на холмистой дороге (на-

⁹Официальный сайт Webots, URL: <https://cyberbotics.com/> (дата обращения: 24.12.2024)

¹⁰Документация Webots, URL: <https://cyberbotics.com/doc/guide/controller-programming> (дата обращения: 07.05.2025)

¹¹Документация Webots, URL: <https://cyberbotics.com/doc/automobile/city> (дата обращения: 07.05.2025)

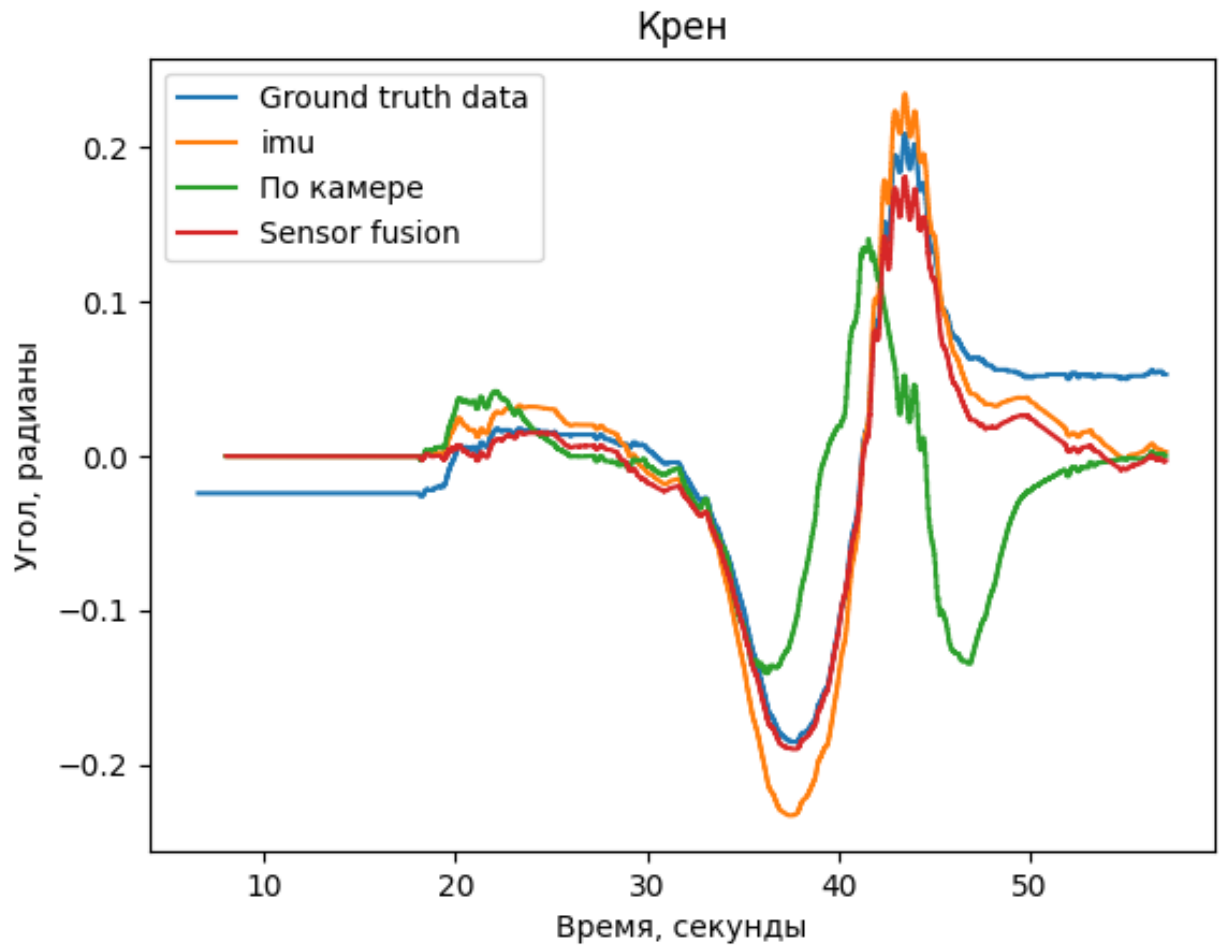


Рис. 10: Сравнение угла тангажа в симуляторе Webots

клоны больше 20 градусов) составила 0.012 радиан (0.69 градусов), а среднеквадратическое отклонение равняется 0.008 радианам (0.46 градусов). В статье «A Novel Roll and Pitch Estimation Approach for a Ground Vehicle Stability Improvement Using a Low Cost IMU»[9] среднее отклонение тангажа угла равняется 0.996 градусам на более ровной поверхности. Численное сравнение не до конца корректно, так как эксперименты проводились на другом симуляторе с другими данными, на которые не приведена ссылка, а код алгоритма, описанного в статье, недоступен. Аналогично корректное сравнение не провести со статьями, найденными на первых страницах по запросу «sensor fusion imu for

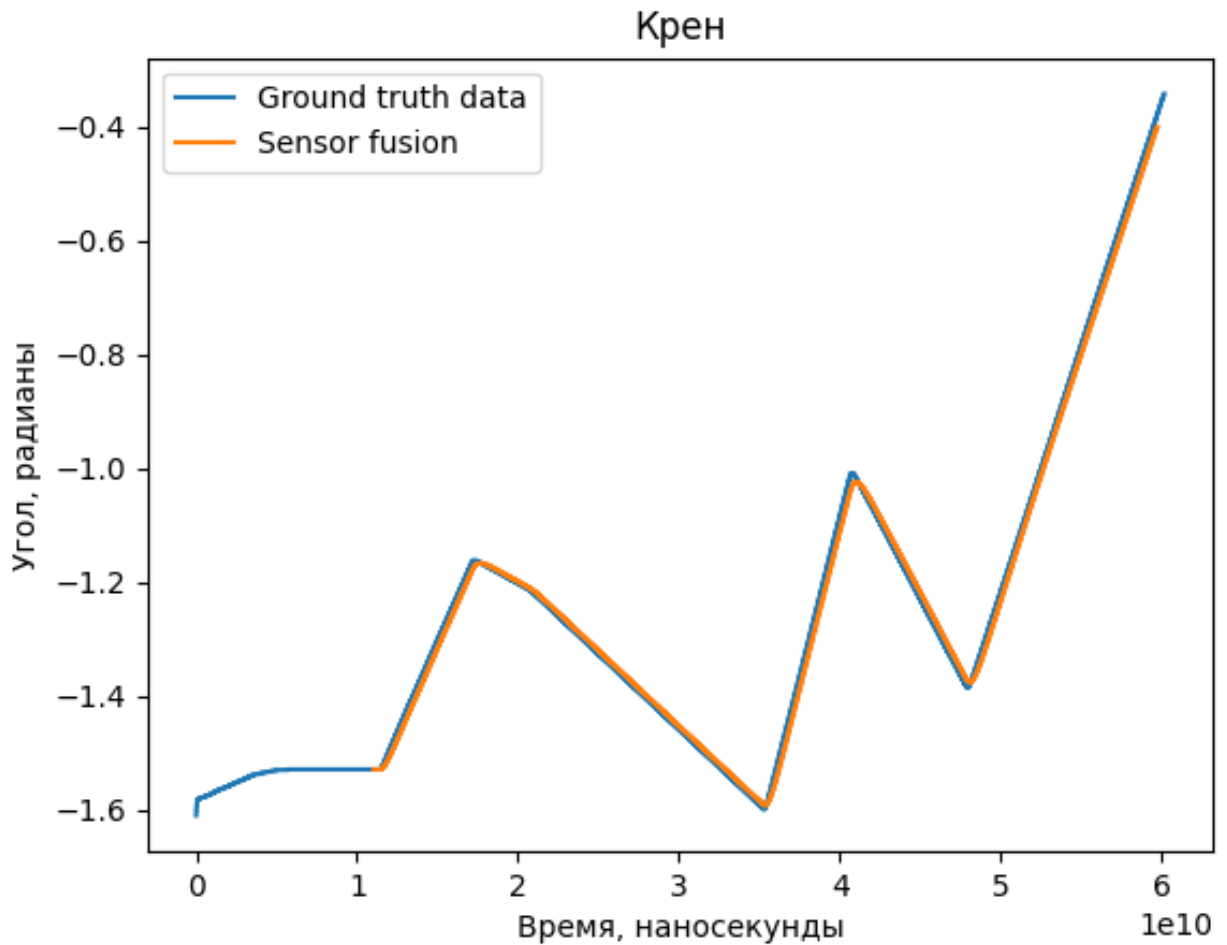


Рис. 11: Сравнение угла крена в симуляторе Webots

orientation estimation» в поисковике Google Scholar¹². Можно сделать предположение, что реализованный алгоритм превосходит по данным метрикам описанный в статье, исходя из того, что перепады высоты составляют большее значение.

Решение	Средняя ошибка	Среднеквадратическое отклонение
Предложенное	0.69°	0.46°
Из статьи	0.996°	0.49°

Средняя ошибка определения угла крена составила 0.52 градуса, среднеквадратическое отклонение равняется 0.4 градусам. В статье

¹²Google Scholar, URL: <https://scholar.google.com/> (дата обращения: 20.05.2025)



Рис. 12: Симулятор Webots, горная дорога

«Accurate Attitude Estimation of a Moving Land Vehicle Using Low-Cost MEMS IMU Sensors»[2] среднееквadraticское отклонение угла составило 0.1 градуса, а средняя ошибка 0.2 градусов, но угол менялся не более, чем на 1 градус по ходу измерений, тогда как в предложенном тестировании угол менялся сильнее.

Решение	Средняя ошибка	Среднееквadraticское отклонение
Предложенное	0.52°	0.4°
Из статьи	0.1°	0.2°

Был совершен круговой проезд длиной более 10 минут, при котором машина вернулась на то же место, откуда стартовала. Было оценено изменение угла рыскания.

По результатам проезда отклонение угла рыскания в конце поездки от начального угла у колесной одометрии составило 2.433 радиан, у IMU — 0.535 радиан, тогда как у результата sensor fusion ошибка составила 0.177 радиан, что значительно меньше. График показан на рис. 13.

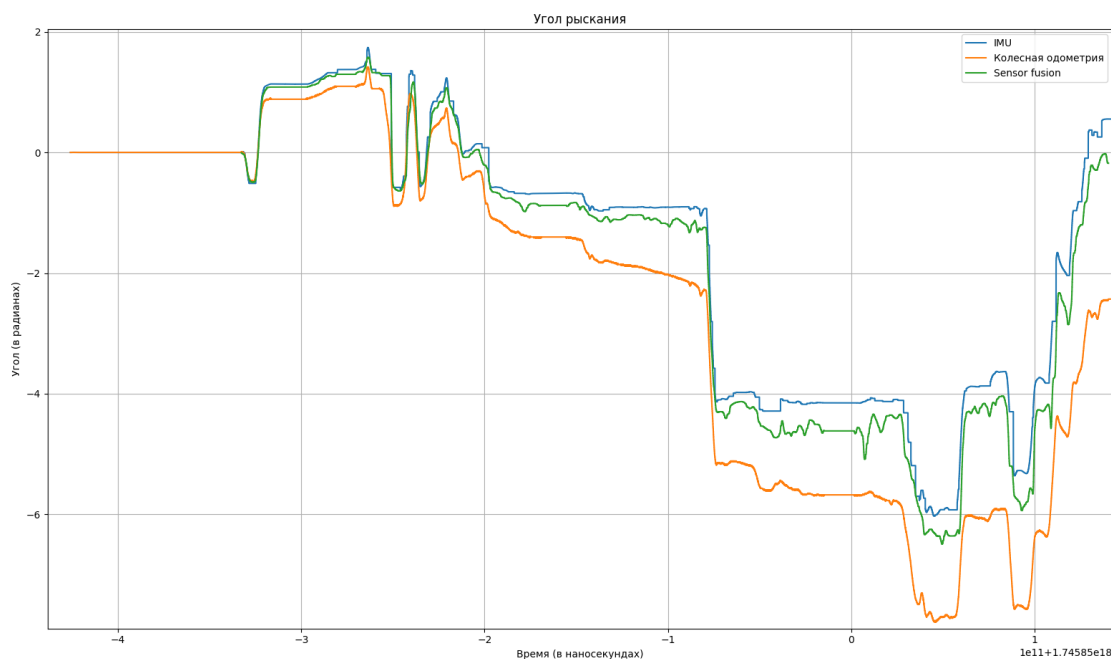


Рис. 13: График изменения угла рыскания при круговом проезде

4.3. Оценка производительности

Была проведена оценка производительности на целевом процессоре ARMv8 Processor rev 1 с архитектурой aarch64. Данный процессор имеет частоту 1510 Гц, вычисления производятся на 4 потоках: данные с каждого датчика обрабатываются в отдельном потоке, а также используется главный поток.

Среднее время работы	Среднеквадратическое отклонение
0.004158 (секунды)	0.003957 (секунды)

Время выполнения алгоритма является удовлетворительным, так как частота алгоритма (240 итераций в секунду без учета издержек на получение данных с датчиков) превышает частоту камеры (30 кадров в секунду).

Заключение

В результате работы были выполнены следующие задачи.

- Был произведен обзор существующих решений.
- Был реализован алгоритм определения ориентации по IMU.
- Был реализован алгоритм определения ориентации по данным с камеры.
- Был реализован алгоритм sensor fusion данных с трех датчиков.
- Был протестирован алгоритм: при помощи симулятора и кругового проезда на автомобиле в Москве, была оценена производительность.

Ссылка на репозиторий не приведена из-за соглашения о неразглашении. В файле на Google Drive¹³ можно увидеть пример оценки изменения горизонта в условиях тряски с помощью реализованного алгоритма по данным с камеры, в другом файле на Google Drive¹⁴ можно увидеть результат проезда в симуляторе Webots, а в третьем файле на Google Drive¹⁵ находится проезд с sensor fusion на реальных данных.

¹³Файл на Google Drive, URL: <https://drive.google.com/file/d/1XjT08jmHUNAhdOGvwBGXWTvrgZwRhQxb/view?usp=sharing> (дата обращения: 24.12.2024)

¹⁴Файл на Google Drive, URL: https://drive.google.com/file/d/1h02uYpkRChwIJle-tS5_gA6D1SjMpaHM/view?usp=sharing (дата обращения: 10.05.2025)

¹⁵Файл на Google Drive, URL: <https://goo.su/pZFgV> (дата обращения: 04.04.2025)

Список литературы

- [1] Aerts Peter, Demeester Eric. [Time Synchronisation of Low-cost Camera Images with IMU Data based on Similar Motion](https://www.scitepress.org/Papers/2019/78376/78376.pdf) // International Conference on Informatics in Control, Automation and Robotics. — 2019. — 01. — P. 292–299. — URL: <https://www.scitepress.org/Papers/2019/78376/78376.pdf> (дата обращения: 2 января 2025 г.).
- [2] Ahmed Hamad, Tahir Muhammad. Accurate Attitude Estimation of a Moving Land Vehicle Using Low-Cost MEMS IMU Sensors // [IEEE Transactions on Intelligent Transportation Systems](#). — 2017. — Vol. 18, no. 7. — P. 1723–1739.
- [3] Kalman R. E. A New Approach to Linear Filtering and Prediction Problems // [Journal of Basic Engineering](#). — 1960. — 03. — Vol. 82, no. 1. — P. 35–45. — https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf.
- [4] Kundra László, Ekler Péter, Charaf Hassan. [Improving orientation estimation in mobiles with built-in camera](#) // 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom). — 2013. — P. 765–770. — URL: <https://ieeexplore.ieee.org/document/6719202> (дата обращения: 2 января 2025 г.).
- [5] Ligorio Gabriele, Sabatini Angelo Maria. Extended Kalman Filter-Based Methods for Pose Estimation Using Visual, Inertial and Magnetic Sensors: Comparative Analysis and Performance Evaluation // [Sensors](#). — 2013. — Vol. 13, no. 2. — P. 1919–1941. — URL: <https://www.mdpi.com/1424-8220/13/2/1919> (дата обращения: 2 января 2025 г.).
- [6] Luciano Buonocore Aerolino De Almeida Neto, Júnior Cairo Lúcio Nascimento. [A sensor data fusion algorithm for indoor slam using a low-cost mobile robot](#) // Adaptive Mobile Robotics. — P. 762–769. —

URL: https://www.worldscientific.com/doi/abs/10.1142/9789814415958_0097 (дата обращения: 2 января 2025 г.).

- [7] A Multi-Sensor Information Fusion Method Based on Factor Graph for Integrated Navigation System / Jing Xu, Gongliu Yang, Yiding Sun, Stjepan Picek // [IEEE Access](#). — 2021. — Vol. 9. — P. 12044–12054. — URL: <https://ieeexplore.ieee.org/document/9324742> (дата обращения: 2 января 2025 г.).
- [8] Nasir Ahmad Kamal, Hille Christof, Roth Hubert. Data Fusion Of Stereo Vision and Gyroscope For Estimation Of Indoor Mobile Robot Orientation // [IFAC Proceedings Volumes](#). — 2012. — Vol. 45, no. 4. — P. 163–168. — 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control. URL: <https://www.sciencedirect.com/science/article/pii/S1474667015404604> (дата обращения: 2 января 2025 г.).
- [9] A Novel Roll and Pitch Estimation Approach for a Ground Vehicle Stability Improvement Using a Low Cost IMU / Malik Kamal Mazhar, Muhammad Jawad Khan, Aamer Iqbal Bhatti, Noman Naseer // [Sensors](#). — 2020. — Vol. 20, no. 2. — URL: <https://www.mdpi.com/1424-8220/20/2/340>.
- [10] Parasuraman S. Sensor Fusion for Mobile Robot Navigation: Fuzzy Associative Memory // [Procedia Engineering](#). — 2012. — Vol. 41. — P. 251–256. — International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012). URL: <https://www.sciencedirect.com/science/article/pii/S1877705812025568> (дата обращения: 2 января 2025 г.).
- [11] Research Polaris Marker. Polaris Marker Research. — 2024. — URL: <https://www.polarismarketresearch.com/industry-analysis/advanced-driver-assistance-systems-adas-market> (дата обращения: 2025-05-20).

- [12] Ribeiro Maria, Ribeiro Isabel. [Kalman and Extended Kalman Filters: Concept, Derivation and Properties](https://doi.org/10.1115/1.3662552). — 2004. — 04. — URL: <https://doi.org/10.1115/1.3662552> (дата обращения: 2 января 2025 г.).
- [13] Sasiadek J., Hartana P. [Sensor data fusion using Kalman filter](https://www.researchgate.net/publication/3859405_Sensor_data_fusion_using_Kalman_filter). — Vol. 2. — 2000. — 08. — P. WED5/19 – WED5/25 vol.2. — URL: https://www.researchgate.net/publication/3859405_Sensor_data_fusion_using_Kalman_filter (дата обращения: 2 января 2025 г.).
- [14] Sensor Data Fusion for a Mobile Robot Using Neural Networks / Andres J. Barreto-Cubero, Alfonso Gómez-Espinosa, Jesús Arturo Escobedo Cabello et al. // [Sensors](https://www.mdpi.com/1424-8220/22/1/305). — 2022. — Vol. 22, no. 1. — URL: <https://www.mdpi.com/1424-8220/22/1/305> (дата обращения: 2 января 2025 г.).
- [15] [Visual gyroscope: Integration of visual information with gyroscope for attitude measurement of mobile platform](https://ieeexplore.ieee.org/document/4694693) / Deokhwa Hong, Hyunki Lee, Hyungsuck Cho et al. // 2008 International Conference on Control, Automation and Systems. — 2008. — P. 503–507. — URL: <https://ieeexplore.ieee.org/document/4694693> (дата обращения: 2 января 2025 г.).
- [16] Wu J. K., Wong Y. F. [Bayesian Approach for Data Fusion in Sensor Networks](https://www.researchgate.net/publication/235892505_A_Bayesian_Approach_to_Data_Fusion_in_Sensor_Networks) // 2006 9th International Conference on Information Fusion. — 2006. — P. 1–5. — URL: https://www.researchgate.net/publication/235892505_A_Bayesian_Approach_to_Data_Fusion_in_Sensor_Networks (дата обращения: 2 января 2025 г.).