

# Уточнение GNSS-позиционирования при помощи визуально-инерциальной одометрии

Карасева Е.О., СПбГУ, Санкт-Петербург liza.1610@mail.ru

## Аннотация

В данной работе предлагается модифицировать визуально-инерциальный SLAM-алгоритм путем добавления в конвейер GNSS-данных. Статья содержит подробный обзор алгоритмов, обоснования и описание работы выбранного алгоритма, а также описание реализации решения.

## Введение

Прогресс в робототехнике ускоряет развитие систем автономной навигации для дронов и автомобилей. Такие системы используют данные с различных датчиков для одновременной локализации и картографирования (SLAM). Популярный подход к решению задачи SLAM — визуально-инерциальная одометрия (VIO), основанная на камерах и данных с инерциальных измерительных модулей (IMU), однако она подвержена накоплению ошибок из-за использования относительных измерений.

Добавление GNSS-данных, представляющих абсолютные координаты, позволяет снизить эти ошибки и повысить точность. В работе предлагается интеграция GNSS в модуль Kimera-VIO с последующей оценкой на расширенном датасете EuRoC. Ожидается повышение точности траектории.

## Решения, интегрирующие GNSS, VIO и SLAM

Интеграция GNSS в VI-SLAM активно исследуется [8], но доступных решений с открытым исходным кодом немного. Чаще всего это модификации существующих визуально-инерциальных алгоритмов с недостаточной документацией и без оценки точности траекторий, что затрудняет их использование. Кроме того, часть из них работает только с GPS, что снижает надёжность и требует доработок для поддержки данных других навигационных систем. Анализ решений представлен в Таблице 1.

Алгоритм	Тип	Обновления	Лицензия
MINS	GNSS	2024	GPLv3
GNSS-Stereo-Inertial Fusion	GNSS	2024	GPLv3

RTK-Visual-Inertial-Navigation	GNSS	2024	GPLv3
InGVIO	GNSS	2023	GPLv3
IC-GVINS	GNSS	2023	GPLv3
SVO-Pro-GPS	GPS	2023	GPLv3
VINS-Fusion	GPS	2021	GPLv3
GVINS	GNSS	2021	GPLv3

Таблица 1: VI-SLAM алгоритмы с поддержкой GNSS и открытым исходным кодом

Ни одно из рассмотренных решений не подходит для использования в коммерческих проектах, поэтому было принято решение провести внедрение GNSS в один из VI-SLAM-алгоритмов самостоятельно.

## Обзор современных реализаций VI-SLAM

В качестве алгоритма для внедрения GNSS-данных было проведено сравнение известных реализаций SLAM с использованием визуально-инерциальной одометрии [2], [3], [4]. Главным критерием отбора являлось наличие открытого исходного кода. Сравнение проводилось по следующим параметрам: тип поддерживаемых камер, метод обработки изображений (прямой или выделения особенностей), способ оценки состояния (вероятностный или на основе оптимизации), наличие механизма коррекции ошибок (замыкание цикла), тип лицензии (необходимая — разрешающая коммерческое использование), а также поддержка проекта — наличие обновлений в течение последних пяти лет.

На основе сравнения VI-SLAM-систем (Таблица 2) можно сделать выводы, что ORB-SLAM3 [9] и Kimera [7] поддерживают все типы камер, VINS-Fusion работает с моно- и стереокамерами, OKVIS ориентирован на стереокамеры, а ROVIO — только на монокулярные. ROVIO существенно отличается благодаря использованию расширенного фильтра Калмана (алгоритма оценки состояния системы на основе нелинейных функций) и прямого метода анализа изображений, что делает его быстрее, но менее точным и более чувствительным к шуму. ORB-SLAM3, VINS-Fusion и Kimera обладают механизмом замыкания циклов для компенсации накопленных ошибок, отсутствующим в OKVIS и ROVIO. Лицензия BSD, подходящая для коммерческого применения, используется в Kimera, OKVIS и ROVIO, тогда как ORB-SLAM3 и VINS-Fusion распространяются под GPLv3. Наибольшую поддержку разработчиков имеют Kimera, ORB-SLAM3 и ROVIO.

Алгоритм	Тип камер	Фронтенд	Бэкенд	Loop closure	Лицензия	Обновления
ORB-SLAM3	Моно, Стерео, RGB-D	Особенности	Оптимизация	Да	GPLv3	2022
VINS-Fusion	Моно, Стерео	Особенности	Оптимизация	Да	GPLv3	2021
OKVIS	Стерео	Особенности	Оптимизация	Нет	BSD 3-Clause	2020
ROVIO	Моно	Прямой метод	Фильтрация	Нет	BSD 3-Clause	2024
Kimera [7]	Моно, Стерео, RGB-D	Особенности	Оптимизация	Да	BSD 2-Clause	2025

Таблица 2: Современные алгоритмы VI-SLAM, данные актуальны на 14.02.25

С учетом указанных преимуществ для интеграции GNSS-данных выбрана библиотека Kimera.

### Диаграмма последовательностей Kimera-VIO

На Рисунке 1 представлена диаграмма последовательностей Kimera-VIO. Библиотека реализует конвейерную параллельность: обработка каждого кадра разбита на этапы, а данные передаются через потокобезопасные очереди и буферы асинхронно.

DataProviderModule отслеживает состояние буферов, собирает пришедшие данные в синхронизированные пакеты и отправляет их во FrontendModule. Он распознает ориентиры и передает их и предсказанную на основе IMU-данных позу в BackendModule, который создает из данных факторы и уточняет позу при помощи оптимизации фактор-графа. MesherModule строит 3D-сетку на основе данных с фронтенда и бекенда, а VisualizerModule готовит данные для визуализации. Наконец, DisplayModule выводит картинку на экран. По окончании чтения датасета DataProviderInterface сигнализирует о завершении, данные перестают поступать в очереди модулей, Pipeline завершает работу каждого из них и возвращает управление в main(), где фиксирует результаты и завершает программу.

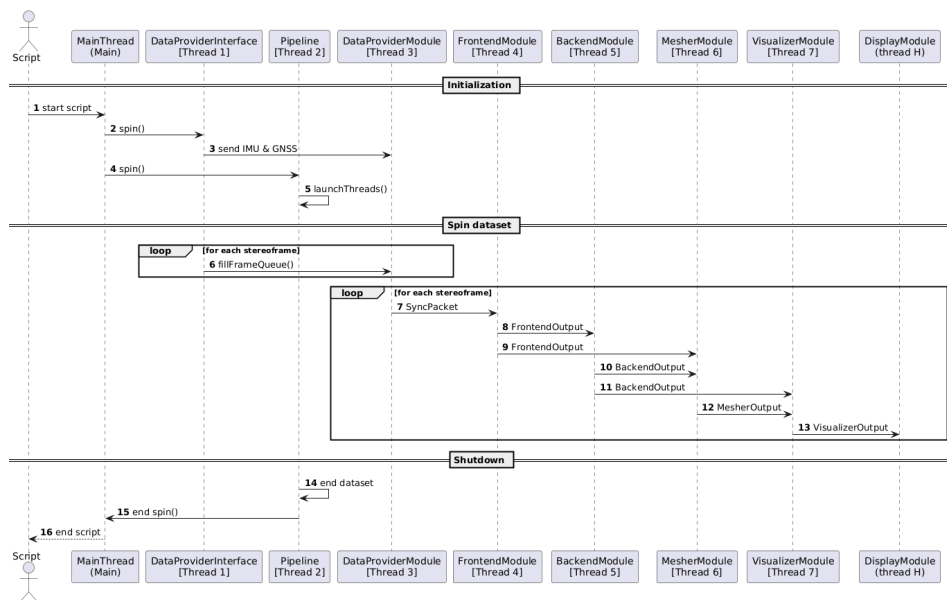


Рис. 1: Диаграмма последовательностей системы Kimera-VIO

## Реализация

Основным требованием к решению является наличие поддержки GNSS-данных в алгоритме и повышение точности построенной траектории при запуске алгоритма на датасете EuRoC. Для выполнения задачи необходимо было разобраться с форматом данных, понять устройство конвейера и модифицировать часть модулей Kimera-VIO [6].

### Формат и преобразования данных

Запуск Kimera-VIO предполагается на датасете EuRoC, содержащем данные с микро летательного аппарата, оснащённого стереокамерами и IMU. В датасете доступны эталонные измерения от системы Vicon (6-мерный вектор состояния с высокой частотой и точностью) и лазерного трекера Leica MS50 (три координаты с точностью до сотых миллиметра). На основе формата этих данных GNSS-измерения также заданы в виде трёх координат (x, y, z) с временной меткой и хранятся в отдельной папке gnss0.

Так как каждый сенсор имеет свою систему координат, перед использованием данные преобразуются в общую систему (в EuRoC это система IMU), с помощью матрицы трансформации  $T_{BS} \in SE(3)$ . Для этого координаты GNSS переводятся в однородные, умножаются на матрицу, а результат преобразуется обратно в аффинные координаты.

### ***Обработка параметров и чтение датасета***

Преобразование выполняется на этапе чтения датасета. Для обработки GNSS-данных был добавлен модуль, отвечающий за чтение и передачу измерений в систему. Матрица TBS, сдвиг по времени и частота измерений хранятся в структуре. Параметры GNSS интегрированы в общую структуру, а сами измерения — в вектор. Передача в следующий модуль идет через callback-функции.

### ***DataProviderModule***

По мере чтения изображений разрозненные данные передаются в модуль, который синхронизирует их по времени. Для GNSS callback-функция помещает измерения в потокобезопасный буфер.

Когда буферы содержат достаточно изображений, IMU и GNSS, из них формируется пакет и передается во фронтенд. Сначала выбирается самый старый кадр левой камеры, затем — интерполируются данные IMU и подбирается ближайшее по времени изображение с правой камеры. Если в GNSS-буфере достаточно данных, проводится линейная интерполяция GNSS-координат по ближайшим временным меткам, после чего результат добавляется к пакету.

### ***FrontendModule***

Фронтенд проводит трекинг ориентиров, триангуляцию и предсказание позы по IMU. Так как фронтенд не использует глобальную позицию, GNSS-данные не влияют на этот этап обработки, а сразу передаются в Backend.

### ***BackendModule***

Бэкенд создает граф факторов [?], на каждой итерации добавляя новые факторы, удаляя устаревшие и вызывая оптимизацию. Результат передаётся

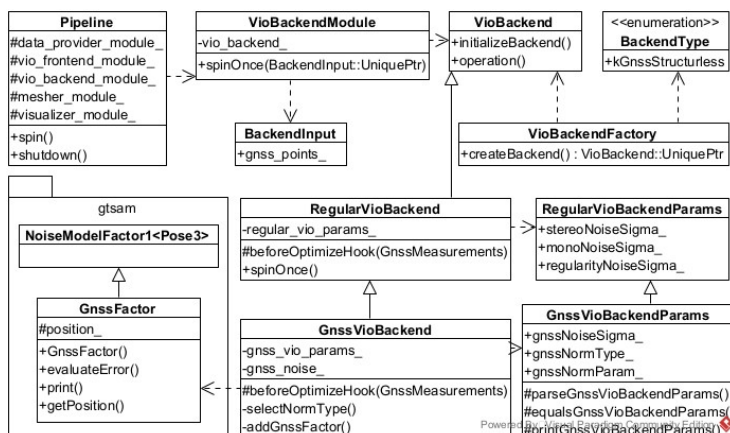


Рис. 2: Архитектура бэкэнд модуля системы

модулям визуализации. В рамках работы был добавлен GNSS-фактор, который учитывает только координаты точек. Тип нормы ошибки (L2, Huber или Tukey) выбирается в параметрах в зависимости от ожидаемого уровня шума. После добавления GNSS-факторы учитываются в общей оптимизации графа (Рисунок 2).

## Эксперимент

Для оценки точности алгоритма использовался модифицированный датасет EuRoC с симулированными GNSS-данными. Анализ траектории проводился на основе файлов журналов, сгенерированных при запуске. Было проведено более сотни запусков алгоритма и подобраны оптимальные параметры. Для эмуляции шума в GNSS-координатах был использован гауссовский шум, это допустимо [1] при упрощенной модели. В Таблице 3 показана полученная статистика: средние результаты реализации с GNSS превосходят стандартную, и в части случаев оказываются лучше значений на лучших запусках, представленных в оригинальной работе [7].

## Заключение

В результате работы был проведен обзор GNSS-VI-SLAM-алгоритмов и VIO-SLAM-систем, потенциально подходящих для внедрения GNSS. После

Location	GNSS				No GNSS	Paper [2]
	Ideal	$\sigma = 0,5\text{м}$	$\sigma = 1,5\text{м}$	$\sigma = 3\text{м}$		
MH_01	<b>0.10</b>	0.27	0.25	0.19	0.16	0.11
MH_02	0.16	0.18	0.29	0.21	0.19	<b>0.10</b>
MH_03	<b>0.16</b>	0.28	0.19	0.23	0.16	0.16
MH_04	0.34	0.44	0.31	0.37	0.28	<b>0.24</b>
MH_05	<b>0.20</b>	0.24	0.22	0.23	0.27	0.35

Таблица 3: Сравнение ошибок построенных траекторий (RMSE ATE)

изучения архитектуры библиотеки Kimera была обеспечена загрузка, синхронизация и передача GNSS-данных в конвейер VIO. Также проведены эксперименты на GNSS-данных с разным уровнем шума: точность построенной траектории увеличилась, алгоритм стабилен даже при редких и неточных GNSS-данных. Реализация представлена по ссылке [10].

Список литературы

[1] Analysis and Discussion on the Optimal Noise Model of Global GNSS Long-Term Coordinate Series Considering Hydrological Loading / Yue-fan He, Guigen Nie, Shuguang Wu, Haiyang Li // Remote Sensing. — 2021. — Vol. 13, no. 3. — URL: <https://www.mdpi.com/2072-4292/13/3/431> (дата обращения: 10 апреля 2025 г.).

[2] Bala Jibril, Adeshina Steve, Aibinu Abiodun. Advances in Visual Simultaneous Localisation and Mapping Techniques for Autonomous Vehicles: A Review // Sensors. — 2022. — 11. — Vol. 22. — P. 8943 (дата обращения: 20 ноября 2024 г.).

[3] A Comprehensive Survey of Visual SLAM Algorithms / Andréa Macario Barros, Yoann Moline, Frédéric Carrel et al. // Robotics. — 2022. — 02. — Vol. 11 (дата обращения: 20 ноября 2024 г.).

[4] A review of visual SLAM for robotics: evolution, properties, and future applications / Basheer Al-Tawil, Thorsten Hempel, Ahmed Abdelrahman, Ayoub Al-Hamadi // Frontiers in Robotics and AI. — 2024. — Vol. Volume 11 - 2024. — URL: <https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1347985> (дата обращения: 12 марта 2025 г.).

- [5] Factor Graphs and GTSAM. — URL: <https://gtsam.org/tutorials/intro.html> (дата обращения: 12 марта 2025 г.).
- [6] Kimera-VIO: Open-Source Visual Inertial Odometry. — URL: <https://github.com/MIT-SPARK/Kimera-VIO> (дата обращения: 20 ноября 2024 г.).
- [7] Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping / Antoni Rosinol, Marcus Abate, Yun Chang, Luca Carlone // CoRR. — 2019. — Vol. abs/1910.02490. — arXiv : 1910.02490 (дата обращения: 20 ноября 2024 г.).
- [8] Nathan Aaron. Loosely Coupled & Tightly Coupled INS & GNSS [2024 Guide]. — 2024. — URL: <https://pointonnav.com/news/loose-vs-tight-coupling-gnss/> (дата обращения: 12 марта 2025 г.).
- [9] ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM / Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez et al. // IEEE Transactions on Robotics. — 2021. — . — Vol. 37, no. 6. — P. 1874–1890. — URL: <http://dx.doi.org/10.1109/TRO.2021.3075644> (дата обращения: 20 ноября 2024 г.).
- [10] Pull-request with GNSS pipeline. — URL: <https://github.com/MIT-SPARK/Kimera-VIO/pull/258> (дата обращения: 20 мая 2025 г.).