

ЯН ГАН, МЭН ВЭЙ-ДА, ДУН ХОУ-ГО, ФЭН НИН-НИН

ИСПОЛЬЗОВАНИЕ КЛЮЧЕВЫХ ТОЧЕК И ЛИНИЙ ИЗОБРАЖЕНИЯ В ЗАДАЧЕ ВИЗУАЛЬНО-ИНЕРЦИАЛЬНОЙ ОДОМЕТРИИ В РЕАЛЬНОМ ВРЕМЕНИ

С целью повышения точности навигации подвижного объекта и надежности извлечения информации из изображений в монокулярных системах на основе методов одновременной локализации и построения карты окружающего пространства (simultaneous localization and mapping – SLAM) при слабо выраженной текстуре сцены предложен метод визуально-инерциальной одометрии с использованием ключевых точек и линий на изображениях. Преимущества метода заключаются в простоте обнаружения линий в реальном пространстве, а также в высокой точности. Комбинирование ключевых точек и линий обеспечивает точное позиционирование SLAM-системы в пространстве со слабо выраженной текстурой, а используемые при этом данные инерциального измерительного модуля (ИИМ) формируют априорную информацию и информацию о масштабе. Оптимизация определения положения производится за счет уменьшения ошибки точечно-линейной репроекции и погрешностей ИИМ методом группового выравнивания точек (bundle adjustment). Предлагается усовершенствованный алгоритм EDlines, который позволяет контролировать длину последовательности пикселей для повышения эффективности распознавания линий и уменьшения их рассогласования. Результаты экспериментов на общедоступных базах данных EuRoC и TUM RGB-D подтверждают работоспособность алгоритма в режиме реального времени, его более высокую точность локализации и робастность по сравнению с визуальными SLAM-методами на основе только ключевых точек и только линий.

Ключевые слова: метод одновременной локализации и построения карты пространства (SLAM-метод), визуально-инерциальная одометрия, ключевые точки, линейные элементы, нелинейная оптимизация.

1. Введение

По мере развития SLAM-методов, а также усовершенствования современных средств машинного зрения устройства на их основе стали находить все более широкое применение в различных областях техники, в частности в беспилотных ле-

Ян Ган. Доктор наук, доцент, Сианьский университет связи и телекоммуникаций (Сиань, КНР).

Мэн Вэй-Да. Студент магистратуры, Сианьский университет связи и телекоммуникаций.

Дун Хоу-Го. Студент магистратуры, Сианьский университет связи и телекоммуникаций.

Фэн Нин-Нин. Студент магистратуры, Сианьский университет связи и телекоммуникаций.

Научный редактор перевода д.т.н. Р.Н. Садеков

тательных аппаратах, автономных транспортных средствах, бытовой робототехнике, системах дополненной и виртуальной реальности. Визуальные SLAM-методы (VSLAM) предполагают применение видеокамер для построения карт окружающего пространства и локализации объектов; их преимуществами являются работа в режиме реального времени и способность генерировать большой объем ценной информации. При этом по сравнению с обычными SLAM-методами, где в качестве основного датчика, как правило, выступает лидар, визуальные SLAM-методы ненадежны и недостаточно точны. В связи с этим исследователи уделяют большое внимание созданию интегрированных систем на основе визуальных SLAM-методов, содержащих датчики колесной одометрии, приемники сигналов спутниковых навигационных систем и ИИМ.

Наибольшим потенциалом развития обладает комбинация систем машинного зрения и ИИМ ввиду ее низкой стоимости. С помощью видеокамер можно получить исчерпывающую информацию об окружающем пространстве, на основе которой могут быть созданы трехмерные модели и осуществляться распознавание собственного местоположения [1]. ИИМ вырабатывает информацию о собственном движении, что позволяет определить масштаб при использовании только монокулярной камеры и получить абсолютную оценку местоположения.

Данные, полученные с помощью средств технического зрения и ИИМ, комплексуются по слабо- или сильносвязанной схеме в зависимости от того, какая именно информация от этих датчиков непосредственно задействуется для оценки местоположения. Сильносвязанные схемы, использующие строгие модели, в которых задействуются данные как одного, так и другого датчиков для общего оценивания местоположения, более популярны благодаря их высокой робастности. Методы комплексирования данных устройств технического зрения и ИИМ можно также разделить на фильтрационные и оптимизационные [2].

Фильтрационные методы, классическим примером которых является обобщенный фильтр Калмана, лучше подходят в условиях ограниченных вычислительных ресурсов и при наличии простых векторов состояния. Оптимизационные методы обладают более высокой точностью, но при этом требуют больше вычислительных ресурсов. Фильтр Калмана с ограничением по нескольким состояниям (Multi-State Constraint Kalman Filter – MSCKF), предложенный коллективом Google, объединяет параметры состояния визуальных систем, формируя сильносвязанную систему машинного зрения и ИИМ с оцениванием в рамках аппарата фильтрации Калмана [3]. Создатели алгоритма ORB-SLAM [4] интегрируют данные ИИМ и визуализации для построения структуры оптимизации графа, что позволяет уменьшить погрешности позиционирования благодаря концепции совместной видимости [5].

Технология VINS-Mono, представленная в [6], демонстрирует большую точность позиционирования по сравнению с другими визуально-инерциальными методами SLAM благодаря слабосвязанной схеме инициализации и сильносвязанной схеме внутренней нелинейной оптимизации [6]. Вместе с тем процесс предварительной обработки данных на основе оптического потока в такой системе зависит от изменений окружающей среды и характеризуется низкой надежностью.

Визуально-инерциальная одометрия демонстрирует значительное улучшение точностных характеристик и надежности по сравнению с визуальной одометрией с применением одного датчика. Методы регистрации движения камеры с учетом фор-

мы ключевых элементов в визуальной одометрии можно разделить на прямые и с использованием ключевых элементов изображений. В соответствии с прямыми методами движение камеры оценивается по изменениям интенсивности изображения, поэтому чрезвычайно важную роль здесь играет освещение, а при слишком быстром движении камеры может произойти сбой отслеживания. Вторая группа методов более надежна с точки зрения устойчивости к изменениям освещения, но в обстановке со слабой текстурой ключевые элементы плохо распознаются, что приводит к прекращению отслеживания и проблемам с оцениванием параметров движения.

В реальных сценариях обстановка со слабой текстурой внутри помещений обычно включает потолки, карнизы, бытовую технику и мебель, которые представляют собой элементы линейного типа. Эти линии меньше зависят от освещения и характеризуются инвариантностью к вращению, а значит, могут служить в качестве уравнивающих ограничений в пространстве со слабой текстурой. Коллектив ученых провел обширные исследования в этой области. Разработчики алгоритма PL-SLAM взяли за основу алгоритм ORB-SLAM и создали монокулярную систему SLAM, функционирующую в режиме реального времени и распознающую одновременно и точки, и линии [7]. Инициализация алгоритма может быть достигнута при выявлении линий всего лишь на трех последовательных изображениях. Тем не менее метод PL-SLAM плохо работает в реальном времени из-за алгоритма распознавания линейного сегмента LSD [8], а кроме того, у него довольно низкая точность.

Метод PL-SVO [9] предусматривает обработку линии в рамках стереовизуальной одометрии, чтобы исключить необходимость непрерывного распознавания и сопоставления элементов на последующих кадрах и обеспечить быстрое отслеживание линейного сегмента. Метод визуально-инерциальной одометрии PL-VIO на основе распознавания ключевых точек и линий [10] позволяет повысить точность и робастность системы за счет добавления линий в процесс оптимизации скользящего окна с помощью алгоритма VINS-Mono. Аналогичным образом работает и PL-VINS. Авторы улучшили алгоритм LSD, скорректировав скрытые параметры и введя процедуру фильтрации линейных элементов, что трехкратно увеличило скорость обработки данных [11].

Определение собственного местоположения по методу PLP-SLAM [12] предполагает привлечение как ключевых точек, так и линий; при этом задействуется модуль сегментирования и восстановления плоскостей для построения структурированной карты с семантическими элементами. Чтобы учесть меняющуюся длину линии, нейросеть может улучшить ее дескриптор путем распознавания ее геометрии и использовать ее в дополнение к точке для более эффективного визуального позиционирования [13]. Метод Manhattan-SLAM [14] заключается в устойчивом отслеживании видеопотока как в манхэттенском, так и в неманхэттенском пространстве для оценивания местоположения с помощью ключевых точек, линий и плоскостей. В модели LGNET [15] применяются нейросети, которые дополняют первоначальные ключевые элементы и объединяют структурные элементы, чтобы сохранить сложные детали. Для интеграции семантического SLAM-метода или моделей с нейросетями со SLAM-системами требуются платформы с достаточным вычислительным ресурсом, например оснащенные встроенными процессорами или сопряженные с графическими процессорами.

Традиционные методы на основе ключевых точек позволяют легко отслеживать сбой, но с их помощью нельзя восстанавливать движение камеры в обстановке со

слабой текстурой. Решение этой проблемы предложено в настоящей статье. Система оптимизирует данные о местоположении методом группового выравнивания, чтобы минимизировать ошибки репроекции точек и линий на последовательных кадрах. По сравнению с традиционным методом, в котором задействуется только ключевая точка, здесь добавляются линии, что позволяет повысить надежность и точность в условиях изменения освещения или в обстановке со слабой текстурой. Основные результаты данной работы заключаются в следующем. Во-первых, усовершенствован алгоритм распознавания линейного сегмента EDlines [16], посредством которого система получает надежные и значимые признаки в виде линий за счет фильтрации линейных элементов. Это сокращает процесс распознавания коротких линейных сегментов и их сопоставления, увеличивает долю их совпадений, а также обеспечивает выполнение процесса отслеживания в режиме реального времени. Во-вторых, сформированная ошибка точечно-линейной репроекции добавляется в алгоритм нелинейной оптимизации на основе группового выравнивания точек с данными ИИМ, в результате объединяется информация от визуальных и инерциальных датчиков.

Статья построена следующим образом. Сначала выполняется проверка эффективности усовершенствованного алгоритма EDlines, затем предложенный метод сравнивается с двумя современными улучшенными методами монокулярной визуально-инерциальной одометрии VI ORB-SLAM и PL-VIO на примере базы данных EuRoC, а также с алгоритмами PL-SLAM и ORB-SLAM на примере базы данных TUM RGB-D. В конце приводятся результаты.

2. Комплексование данных визуально-инерциальной одометрии и линий на изображении

Описываемая в настоящей статье система VI-SLAM отслеживает движение устройств относительно друг друга и их местоположение, а их ориентация в земной инерциальной системе координат $(\cdot)^E$ остается неопределенной. В связи с этим система координат первого ключевого кадра, определяемого VI-SLAM, как правило, принимается в качестве мировой системы координат $(\cdot)^W$, оси z которой соответствует в ней направлению ускорения силы тяжести. Система координат $(\cdot)^B$ связана с корпусом объекта (далее – связанная система координат) и совпадает с системой координат ИИМ. Система координат камеры обозначается как $(\cdot)^C$. Повороты представлены с помощью матриц вращения R или кватернионов q . Точка в мировой системе координат или в реальном трехмерном пространстве обозначается как \mathbb{R}^3 , а точка на плоскости изображения – как \mathbb{R}^2 . Вектор ускорения силы тяжести в мировой системе координат представлен в виде $g^w = [0 \ 0 \ g]^T$, а его направление соответствует оси z . Если вектор описывает относительное преобразование одной системы координат в другую, нижний индекс справа указывает на преобразуемую систему координат. Например, P_{CB} обозначает вектор преобразования системы координат камеры в связанную систему координат ИИМ. Что касается матрицы вращения, то преобразование ориентации из системы координат камеры в систему координат ИИМ обозначается как R_{CB} . Камера и ИИМ жестко закреплены на одном кронштейне. Преобразование $T_{CB} = [R_{CB}|P_{CB}]$ (4×4) между системой координат камеры $(\cdot)^C$ и связанной системой координат ИИМ $(\cdot)^B$ обычно неизвестно и требует калибровки.

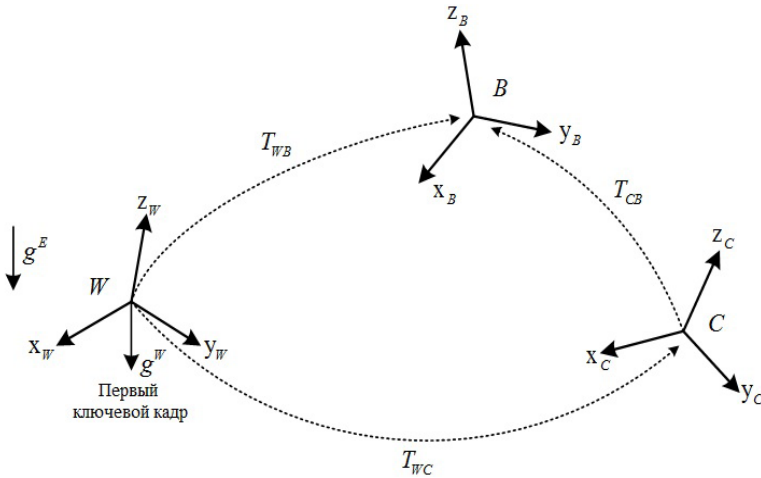


Рис. 1. Взаимосвязь мировой системы координат $(\cdot)^W$, системы координат камеры $(\cdot)^C$ и связанной системы координат ИИМ $(\cdot)^B$. Параметры g_E и g_W – векторы ускорения силы тяжести в земной инерциальной системе координат $(\cdot)^E$ и мировой системе координат $(\cdot)^W$ соответственно

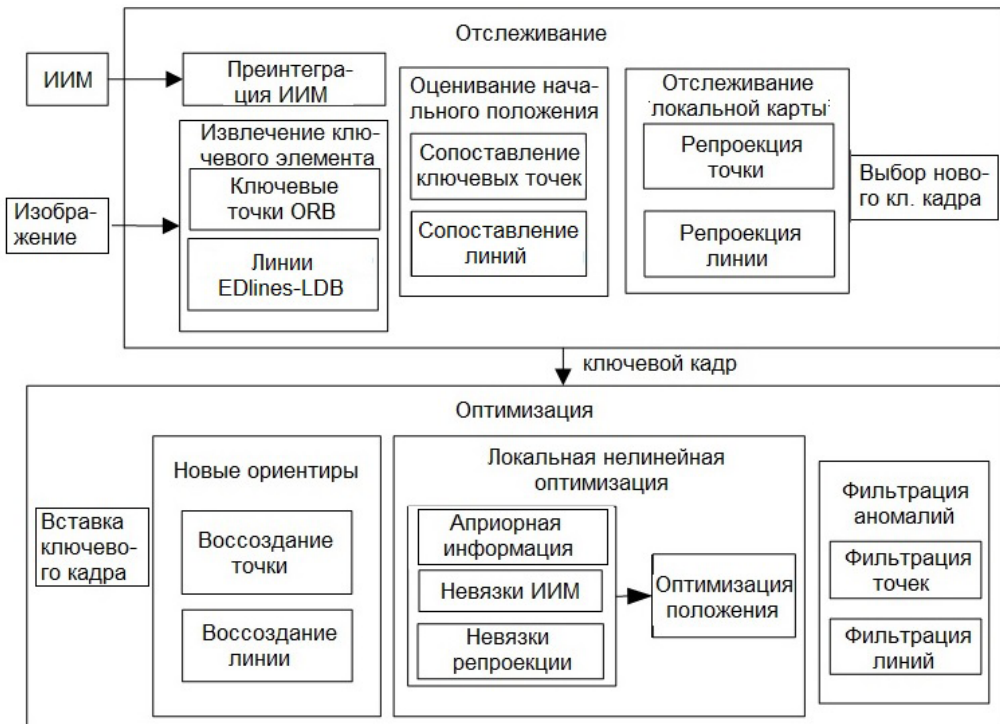


Рис. 2. Блок-схема нелинейной оптимизированной визуально-инерциальной одометрии на основе ключевых точек и линий

2.1 Структура алгоритма

Как видно из рис. 2, предложенный алгоритм состоит из двух основных этапов – слежения и оптимизации. Входными данными для канала слежения являются изображения, полученные от монокулярной камеры, и показания ИИМ. Из каждо-

го изображения с помощью ORB-детектора извлекаются ключевые точки, а посредством алгоритма EDlines – линии. Последние сопоставляются с использованием дескрипторов типа LBD (Line Band Descriptor). Одновременно с этим производится предварительная обработка данных ИИМ [17, 18] для определения начального местоположения и скорости в текущей системе координат. На этом этапе происходит перепроецирование ключевых точек и линий на изображении для построения локальной карты. После формирования новых ключевых кадров поток отслеживания передает их в виде исходных данных в поток оптимизации. Принятые ключевые кадры сопоставляются с локальной картой для построения ориентиров. Наблюдаемое положение ориентиров комбинируется с ранее известной информацией, тем самым осуществляется локальная оптимизация методом группового уравнивания точек для коррекции ошибки точно-линейной репроекции и погрешности предварительной обработки данных ИИМ, чтобы уточнить местоположение и удалить ошибочные ориентиры. Таким образом, выполняется оптимизационная часть алгоритма.

2.2 Предварительная обработка данных ИИМ

ИИМ обычно содержит триады гироскопов и акселерометров, которые измеряют угловые скорости и ускорения датчиков в инерциальной системе координат. Модель измерений выглядит следующим образом:

$$\tilde{\omega}_{WB}^B = \omega_{WB}^B + b^g + \eta^g, \quad (1)$$

$$\tilde{a}^B = R_{WB} (a^W - g^W) + b^a + \eta^a, \quad (2)$$

где $\tilde{\omega}_{WB}^B$ – измеренное значение угловой скорости, \tilde{a}^B – измеренное значение ускорения, b^g – смещение нуля гироскопа, b^a – смещение нуля акселерометра, η^g и η^a – белый шум датчиков, g^W – ускорение силы тяжести в мировой системе координат.

Концепция предварительной обработки данных ИИМ впервые была предложена в статье [17] как метод работы в евклидовом пространстве. В развитие этой идеи в [18] данный подход был применен к локально-евклидову пространству, в результате чего было получено приращение относительного движения независимо от первоначального состояния системы. При наличии двух последовательных кадров и измерений ИИМ в моменты времени i и j можно рассчитать ориентацию R_{WBj} , местоположение P_{WBj} и скорость V_{WBj} по приведенной ниже формуле, описывающей множественные значения измерений за период:

$$\begin{aligned} R_{WBj} &= R_{WBi} \prod_{k=i}^{j-1} \text{Exp}\left(\left(\omega_k^B - b_k^g - \eta_k^g\right) \Delta t\right), \\ P_{WBj} &= P_{WBi} + \sum_{k=i}^{j-1} \left(V_{WBk} \Delta t + \frac{1}{2} g^W \Delta t^2 + \frac{1}{2} R_{WBk} \left(a_k^B - b_k^a - \eta_k^a \right) \Delta t^2 \right), \\ V_{WBj} &= V_{WBi} + g^W \Delta t_{ij} + \sum_{k=i}^{j-1} R_{WBk} \left(a_k^B - b_k^a - \eta_k^a \right) \Delta t, \end{aligned} \quad (3)$$

где Δt – интервал выборки данных ИИМ, а $\Delta t_{ij} = (j - i)\Delta t$. Параметр $\text{Exp}(\cdot)$ представляет собой оператор экспоненциального отображения группы алгебры Ли $\mathfrak{so}(3)$.

В оригинале статьи авторы употребляют термин «преинтеграция».

Если пренебречь влиянием измерительного шума ИИМ, с учетом смещения нуля $\bar{b}_i^{(c)}$ в момент времени i приращение смещения нуля в момент времени j составляет $\delta b_i^{(c)}$. Выражение в (3) можно переписать следующим образом:

$$\begin{aligned} R_{WBj} &= R_{WBi} \Delta \bar{R}_{ij} \text{Exp}(J_{\Delta \bar{R}_{ij}}^g \delta b_i^g), \\ P_{WBj} &= P_{WBi} + V_{WBi} \Delta t_{ij} + \frac{1}{2} g^W \Delta t_{ij}^2 + R_{WBi} (\Delta \bar{P}_{ij} + J_{\Delta \bar{P}_{ij}}^g \delta b_i^g + J_{\Delta \bar{P}_{ij}}^a \delta b_i^a), \\ V_{WBj} &= V_{WBi} + g^W \Delta t_{ij} + R_{WBi} (\Delta \bar{V}_{ij} + J_{\Delta \bar{V}_{ij}}^g \delta b_i^g + J_{\Delta \bar{V}_{ij}}^a \delta b_i^a), \end{aligned} \quad (4)$$

где функциональные определители Якоби $J_{(c)}^g$ и $J_{(c)}^a$ описывают характер изменений в измерениях, связанных с изменениями в оценке смещения нуля [18]. Значения смещения нуля \bar{b}_i^g , \bar{b}_i^a остаются постоянными в течение периода предварительной обработки данных ИИМ, параметры которого $\Delta \bar{R}_{ij}$, $\Delta \bar{P}_{ij}$, $\Delta \bar{V}_{ij}$ не зависят от состояния и силы тяжести в момент времени i и могут быть рассчитаны непосредственно по выходным данным ИИМ между двумя ключевыми кадрами, а именно:

$$\begin{aligned} \Delta \bar{R}_{ij} &= \prod_{k=i}^{j-1} \text{Exp}((\omega_k^B - \bar{b}_i^g) \Delta t), \\ \Delta \bar{P}_{ij} &= \sum_{k=i}^{j-1} \left(\Delta \bar{V}_{ik} \Delta t + \frac{1}{2} \Delta \bar{R}_{ik} (a_k^B - \bar{b}_i^a) \Delta t^2 \right), \\ \Delta \bar{V}_{ij} &= \sum_{k=i}^{j-1} \Delta \bar{R}_{ik} (a_k^B - \bar{b}_i^a) \Delta t. \end{aligned} \quad (5)$$

2.3 Распознавание ключевых точек и линий

Для распознавания ключевых точек обычно используются ORB-признаки, а в визуальных SLAM-методах для обнаружения и распознавания линий чаще всего применяется алгоритм LSD, определяющий градиенты каждого пикселя и объединяющий пиксели. Зависимость от градиентов и процедура объединения пикселей требуют значительных вычислительных ресурсов, что сказывается на способности визуального SLAM-метода работать в режиме реального времени. Для быстрого распознавания линий авторы данной статьи привлекают более эффективный алгоритм EDlines. Сопоставление линий производится с помощью дескриптора LBD. Процесс распознавания ключевых точек и линий показан на рис. 3.

Сначала устанавливается порог распознавания и выделяются пиксельные точки с уровнем интенсивности больше или меньше порогового значения на изображении, чтобы реализовать его бинаризацию и преобразовать в монохромное бинарное изображение для следующего этапа обработки. Извлечение ключевых элементов в различных масштабах выполняется с помощью N-слойной гауссовой пирамиды, а их анализ и обработка в различных разрешениях – с помощью N-масштабной пространственной пирамиды. Использование обеих пирамид позволяет провести более полный анализ изображения и решить такие задачи, как обнаружение объекта, его распознавание и сопоставление его изображений в различных масштабах [19]. На каждом слое выделяют особые (ключевые) точки с помощью детектора Oriented FAST. Затем посредством алгоритма дерева квадрантов особые точки, выявленные детектором Oriented FAST, равномерно распределяются в каждом слое пирамиды.

После этого алгоритм дерева квадрантов запускается еще раз для исходного изображения, чтобы получить равномерное и рассеянное распределение особых точек и одновременно сократить дублирование.

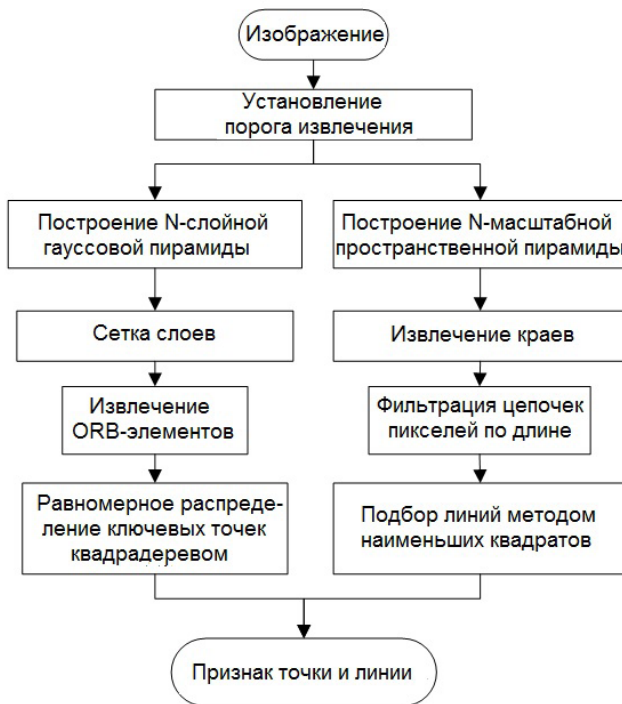


Рис. 3. Блок-схема процесса выделения ключевых точек и линий

2.4 Обнаружение линии

При анализе сложных изображений традиционный алгоритм обнаружения линии зачастую позволяет находить большое количество коротких линейных сегментов. В случае слабых текстур при недостаточных ограничениях визуального SLAM-метода могут фиксироваться нагромождения линий, что повышает вычислительную нагрузку при их обнаружении, распознавании, описании и сопоставлении. Как следствие, увеличивается вероятность неправильного сопоставления линий. Мы ввели в исходный алгоритм распознавания линейных сегментов EDlines этап фильтрации длины для исключения линейных сегментов, длина которых меньше установленного порога. Ключевые элементы большей длины будут обнаружены и распознаны на серии изображений с большей вероятностью, что обеспечит более высокую точность позиционирования. Задача этого алгоритма – создать четкие, непрерывные и достаточно длинные фрагменты контуров. При обработке черно-белого изображения усовершенствованный алгоритм EDlines состоит из нескольких этапов.

1. Извлечение краев. Сначала исходное изображение подвергается гауссовой фильтрации (с использованием гауссова ядра в виде матрицы 5×5 с дисперсией 1 по умолчанию). Затем выполняется расчет величин и направлений градиентов по следующим формулам:

$$\begin{aligned}
 g_x(x, y) &= \frac{L(x+1, y) - L(x, y) + L(x+1, y+1) - L(x, y+1)}{2}, \\
 g_y(x, y) &= \frac{L(x, y+1) - L(x, y) + L(x+1, y+1) - L(x+1, y)}{2}, \\
 g(x, y) &= \sqrt{g_x(x, y)^2 + g_y(x, y)^2}, \theta = \arctan \frac{g_x(x, y)}{-g_y(x, y)}.
 \end{aligned}
 \tag{6}$$

Здесь $g(x, y)$ и θ обозначают соответственно величину и направление градиента. Пиксели выбираются в качестве точек привязки, если их градиент выше, чем у соседних пикселей в том же направлении, и не менее μ . В конце этого этапа точки привязки соединяются и формируют первичные цепочки пикселей (контуры).

2. Фильтрация по длине. Первичные цепочки пикселей фильтруются по длине, то есть из обработки исключаются цепочки, длина которых меньше установленного порога. Задача данного этапа – обеспечить, чтобы извлеченные ключевые элементы состояли в основном из длинных линейных сегментов. Представим конечные точки цепочки пикселей как $a(x_a, y_a)$ и $b(x_b, y_b)$

$$\begin{cases} len_n = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \\ len_n \geq len_{min} = \sigma \min(W, H) \end{cases}
 \tag{7}$$

где len_{min} – минимальная длина цепочки пикселей, W и H – ширина и высота изображения, σ – масштабный коэффициент длины.

3. Подбор линий методом наименьших квадратов. Цепочки пикселей, полученные с помощью алгоритма извлечения краев, объединяются в один или несколько линейных сегментов. После этого каждый сегмент подбирается методом наименьших квадратов. Если погрешность превышает определенное пороговое значение, формируется новый линейный сегмент. Процесс повторяется для всех остальных пикселей цепочки до тех пор, пока не будет выполнена обработка всех точек привязки:

$$\min \sum_{a=1}^m |\delta_a| = \sum_{a=1}^m |\varphi(x_a) - y_a|,
 \tag{8}$$

где x_a и y_a – координаты пикселя на конце соседних линейных сегментов.

2.5 Представление ключевых точек и линий и модель ошибки репроекции

В трехмерном пространстве линия может быть представлена с помощью координат Плюкера, то есть шестимерного вектора $L = (m^T, d^T)^T \in \mathbb{R}^6$, где $m \in \mathbb{R}^3$ – обычный вектор, $d \in \mathbb{R}^3$ – вектор момента и оба они удовлетворяют условию $n^T d = 0$. Предположим, у нас есть линия L_w в мировой системе координат. Ее можно преобразовать в линию L_c в системе координат камеры с помощью информационной матрицы

$$T_{CW} = \begin{bmatrix} R_{CW} & t_{CW} \\ 0 & 1 \end{bmatrix}.
 \tag{9}$$

Здесь R_{CW} (3×3) и t_{CW} (3×1) представляют собой матрицы поворота и перемещения (или внешних параметров), которые преобразуются из мировой системы координат

в систему координат камеры. Проекцию линии на плоскость камеры можно описать уравнением проекции

$$L_C = \begin{bmatrix} m_C \\ d_C \end{bmatrix} = T_{CW} L_{CW} = \begin{bmatrix} R_{CW} & [t_{CW}] \times R_{CW} \\ 0 & R_{CW} \end{bmatrix} \begin{bmatrix} m_W \\ d_W \end{bmatrix}. \quad (10)$$

Как показано на рис. 4, если новый линейный сегмент присутствует на двух разных изображениях, его представляют с помощью координат Пюкера. На рис. 4, б отмечена трехмерная линия L , которая обозначена как $c1$ и $c2$ в двух полях зрения камеры в разные моменты времени – z_L^{c1} и z_L^{c2} соответственно.

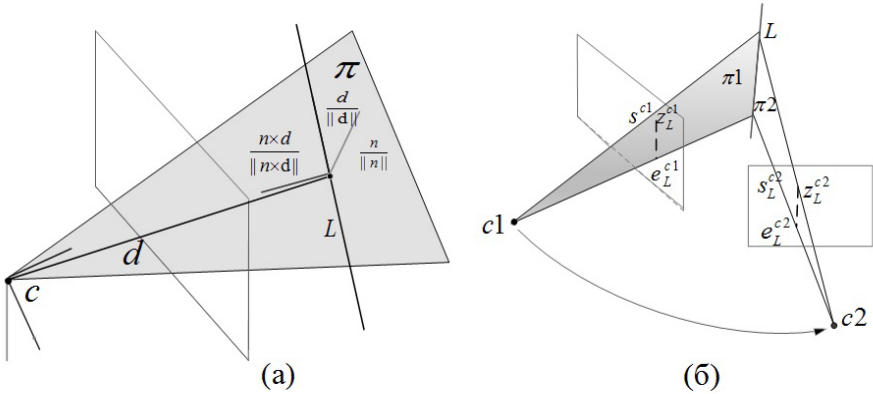


Рис. 4. Координаты Пюкера (а), модель погрешности репроекции линейного сегмента (б)

Предполагается, что внешняя матрица T_{CB} , которая представляет преобразование системы координат камеры в систему координат ИИМ, известна. Вектор состояния кадра изображения i обозначен как $\{R_{WB_i}^R, P_{WB_i}^R, V_{WB_i}^R, b_i^g, b_i^a\}$, а наблюдаемое положение ключевого элемента k в кадре изображения – как \hat{p}_{ik} . Неопределенность местоположения ключевого элемента принята за один пиксель. Таким образом, погрешность репроекции ключевой точки изображения может быть вычислена следующим образом:

$$e_{ik}^p = \hat{p}_{ik} - \frac{1}{z_C} K [I_3 \quad 0_{3 \times 1}] \begin{bmatrix} \ell_k^C \\ 1 \end{bmatrix} = \hat{p}_{ik} - \frac{1}{z_C} K [I_3 \quad 0_{3 \times 1}] T_{CB} \begin{bmatrix} R_{WB_i} & P_{WB_i} \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \ell_k \\ 1 \end{bmatrix}, \quad (11)$$

где K – характеристическая матрица камеры, z_C – третий компонент координат ключевого элемента ℓ_k^C в системе координат камеры, удовлетворяющий условию

$$\ell_k^C = R_{CB} R_{WB_i}^{-1} (\ell_k - p_i) + t_{CB}. \quad (12)$$

Проекцию l_C ключевого элемента L_C на плоскость изображения можно представить в координатах Пюкера:

$$l_C = K_L m_C = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} m_C = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}. \quad (13)$$

Здесь f_x, f_y и c_x, c_y – фокусные расстояния и координаты принципиальной точки камеры соответственно, K_L – характеристическая матрица, которая используется для

проецирования трехмерной линии на плоскость изображения. Как видно из рис. 4, б, погрешность репроекции трехмерного линейного сегмента на двухмерную плоскость текущего кадра изображения можно определить следующим образом:

$$e_{ik}^l = \left[\begin{array}{c} \frac{x_s^T l_c}{\sqrt{l_1^2 + l_2^2}} \quad \frac{x_e^T l_c}{\sqrt{l_1^2 + l_2^2}} \end{array} \right]^T, \quad (14)$$

где $\{x_s, x_e\}$ – двумерные начальная и конечная точки распознанного линейного сегмента.

2.6 Сильносвязанные системы визуально-инерциальной одометрии с использованием ключевых точек и линий

Чтобы оптимизировать положение камеры или 3D-карт за счет минимизации погрешности репроекции на плоскость изображения, визуальные SLAM-методы предполагают применение группового уравнивания точек (bundle adjustment). Этот нелинейный процесс оптимизации можно представить с помощью схемы, приведенной на рис. 5 [20, 21]. Круги на рис. 5, а обозначают положения камеры или ключевые точки, которые необходимо оптимизировать, а квадраты – визуальные наблюдения, которые служат ограничениями между узлами. Помимо ключевых точек в качестве ограничений используются также линии. В случае визуально-инерциальных SLAM-методов используется сильносвязанная структура для оптимизации всех переменных состояния, как показано на рис. 6, б. Кроме того, в схему добавлена предварительная обработка данных ИИМ для ограничения состояний последнего в течение длительных периодов времени.

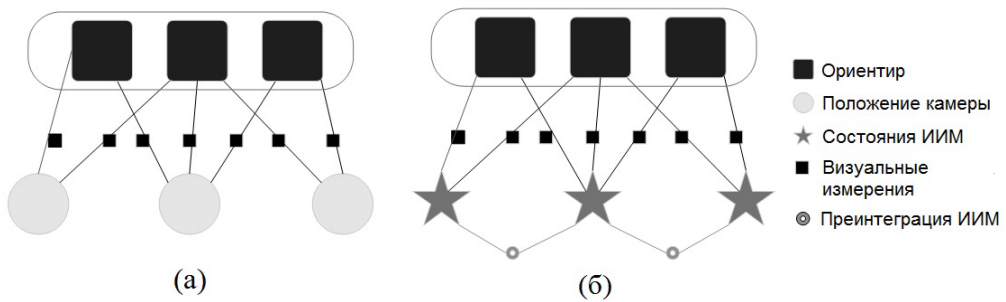


Рис. 5. Схема ограничивающих факторов для визуального SLAM-метода (а), схема ограничивающих факторов для визуально-инерциального SLAM-метода (б)

Этап оптимизации совместной обработки данных играет ведущую роль в методе расчета общего пройденного расстояния. Главным новшеством, предложенным в статье, на этом этапе является привлечение линий в качестве эффективных ограничивающих условий для компенсации недостатка особых точек в обстановке со слабой текстурой. Кроме того, процесс оптимизации на каждой итерации выполняется с применением модели погрешности точно-линейной репроекции. Состояние, полученное в результате оптимизации, далее служит априорной информацией для улучшения последующих кадров и опорным параметром для уменьшения вычислительной нагрузки и достижения более качественных результатов.

После оценивания положения камеры выполняется проецирование точек и линий на локальную карту и сопоставление ключевых точек и линий. Затем производится оптимизация текущего кадра j путем уменьшения погрешности репроекции для всех совпавших точек и линий, а также погрешностей ИИМ. В процессе отслеживания непосредственно после обновления карты вектор погрешностей ИИМ соединяет текущий кадр j с последним ключевым кадром i . Оценка состояния кадра j задается в виде $\theta = \{R_{WBj}, P_{WBj}, V_{WBj}, b_j^g, b_j^a\}$:

$$\theta^* = \arg \min_{\theta} \left(\sum_k E_{proj}(k, j) + E_{IMU}(i, j) \right). \quad (15)$$

Здесь E_{proj} – функция погрешности точно-линейной репроекции:

$$E_{proj}(k, j) = \rho((x^k - \pi(\ell_k^C) \sum_k (x^k - \pi(\ell_k^C))) + \rho((x^k - \pi(L_C)) \sum_k (x^k - \pi(L_C))), \quad (16)$$

где \sum_k – информационная матрица, связанная с масштабом ключевых точек.

Вектор погрешностей ИИМ $E_{ИИМ}$ определяется следующим образом:

$$E_{ИИМ}(i, j) = \rho \left(\left[\dot{e}_R \dot{e}_V \dot{e}_P \right] \sum_I \left[\dot{e}_R \dot{e}_V \dot{e}_P \right] \right) + \rho \left(\dot{e}_b \sum_R e_b \right), \quad (17)$$

где \sum_I – информационная матрица измерений ИИМ, \sum_R – информационная матрица случайного ухода, ρ – функция стоимости Хубера [18]. Для решения описанной задачи оптимизации применяется алгоритм Гаусса–Ньютона, реализованный в g2o [22].

После оптимизации полученные оценка состояния и матрица Гесса принимаются за априорные данные для следующего шага оптимизации. В случае отсутствия обновлений карты следующий кадр $j + 1$ будет оптимизироваться за счет внедрения оценки состояния из кадра j и использования значения, вычисленного в конце предыдущего шага оптимизации:

$$\theta = \{R_{WBj}, P_{WBj}, V_{WBj}, b_j^g, b_j^a, R_{WBj+1}, P_{WBj+1}, V_{WBj+1}, b_{j+1}^g, b_{j+1}^a\}, \quad (18)$$

$$\theta^* = \arg \min_{\theta} \left(\sum_k E_{proj}(k, j+1) + E_{IMU}(j, j+1) + E_{prior}(j) \right);$$

$$E_{prior}(j) = \rho \left(\left[e_R^T e_V^T e_P^T e_b^T \right] \sum_p \left[e_R^T e_V^T e_P^T e_b^T \right]^T \right); \quad (19)$$

$$e_R = \text{Log}(\bar{R}_{WBj} R_{WBj}) \quad e_V = \bar{V}_{WBj} - V_{WBj},$$

$$e_P = \bar{P}_{WBj} - P_{WBj} \quad e_b = \bar{b}_j - b_j. \quad (20)$$

В приведенном выше уравнении (\cdot) и \sum_p – расчетное состояние и матрица Гесса, сформированные на предыдущем этапе. После прохождения этого этапа оптимизации кадр j станет учтенным [23]. Алгоритм использует априорные данные и непрерывно соединяет последовательные оптимизации кадров до тех пор, пока на карте не появится изменение. В этот момент предыдущий кадр становится недействительным, и процесс отслеживания снова соединит текущий кадр с последним ключевым для продолжения оптимизации.

3. Экспериментальная проверка и анализ результатов

Цель исследования – подтвердить работоспособность предложенного комбинированного метода на основе визуально-инерциальной одометрии с использованием

линий для повышения точности и устойчивости локализации в обстановке со слабой текстурой по сравнению с традиционной визуальной одометрией по ключевым точкам. Для этого задействуются две базы данных: EuRoC [24] и TUM RGB-D [25]. Все эксперименты проводились на компьютере с процессором AMD 4800HS (2,90 ГГц), объемом оперативной памяти 16 ГБ и операционной системой Ubuntu 16.04.

3.1 Эксперимент с алгоритмом распознавания линейного сегмента

Чтобы оценить работоспособность предложенного алгоритма распознавания линии и определить его эффективность с точки зрения усовершенствования системы визуального SLAM-метода, из базы данных EuRoC было выбрано 5 эпизодов для трех экспериментов: MH_03_medium, V1_02_medium, MH_05_difficult, V1_03_difficult и V2_03_difficult. База данных EuRoC состоит из 11 последовательностей, которые относятся к простому, среднему и высокому уровням сложности в зависимости от различных значимых факторов [24]. Характеристики и уровни сложности эпизодов из базы данных EuRoC представлены в табл. 1.

Таблица 1

Наименование	Длина/продолжительность	Примечания
V1_01_easy (простой)	58,6 м/144 с	Медленное движение, яркое изображение
V1_02_medium (средний)	75,9 м/83,5 с	Быстрое движение, яркое изображение
V1_03_difficult (сложный)	79,0 м/105 с	Быстрое движение, размытое изображение
V2_01_easy (простой)	36,5 м/112 с	Медленное движение, яркое изображение
V2_02_medium (средний)	83,2 м/115 с	Быстрое движение, яркое изображение
V2_03_difficult (сложный)	86,1 м/115 с	Быстрое движение, размытое изображение
MH_01_easy (простой)	80,6 м/182 с	Хорошая текстура, яркое изображение
MH_02_easy (простой)	73,5 м/150 с	Хорошая текстура, яркое изображение
MH_03_medium (средний)	130,9 м/132 с	Быстрое движение, яркое изображение
MH_04_difficult (сложный)	91,7 м/99 с	Быстрое движение, темное изображение
MH_05_difficult (сложный)	97,6 м/111 с	Быстрое движение, темное изображение

После расчета градиента в ED в первую очередь устанавливается пороговое значение μ для исключения пикселей, которые не содержат элементы контура (см. раздел 2.3). Максимальная погрешность дискретизации двух последовательных пикселей со-

ставляет 2 и наблюдается в случае, когда ошибки соседних пикселей составляют +1 и -1. При значении углового допуска, равном $22,5^\circ$, порог градиента рассчитывается следующим образом:

$$\mu = \frac{1}{\sin(22,5)} = 5,22.$$

Для экспериментов выбирают пороговое значение для фильтрации длины. Согласно (7), размер выходного изображения из двух общедоступных баз данных, задействованных в экспериментах, составляет 752×480 пикселей (Ш \times В). Предложенный алгоритм применялся в серии экспериментов с последовательностью MN_05_difficult, результаты которых представлены на рис. 6. Поскольку σ непосредственно влияет на описание идентифицированного ключевого элемента и на время сопоставления, чем больше будет значение σ , тем меньше линейных сегментов будет распознано. С точки зрения эффективности наилучший интервал значений σ – 0,07 ... 0,12. В описываемом эксперименте заданная величина σ , при которой было достигнуто большинство наилучших результатов, составляет 0,09.

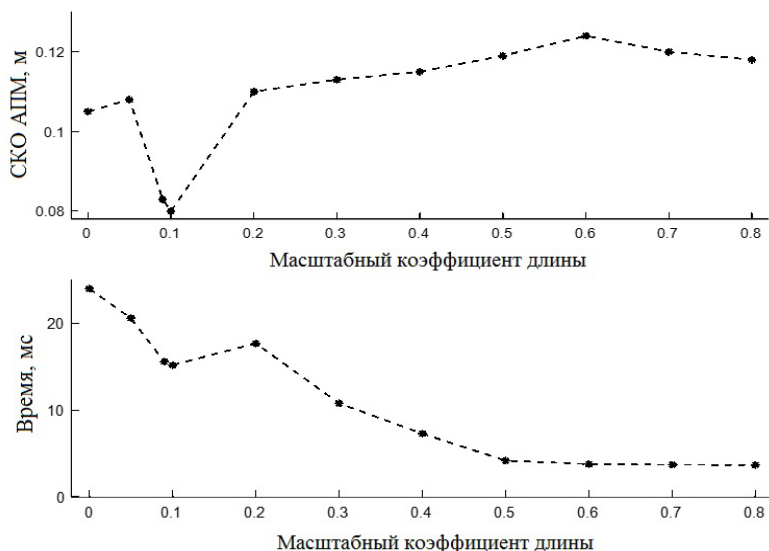


Рис. 6. Исследование фактора фильтрации длины. Зависимость среднего времени отслеживания линии и средней абсолютной погрешности местоположения (АПМ) от различных значений масштабного коэффициента длины для последовательностей MN_05

Получив наилучшие характеристики, мы сравнили алгоритмы LSD, EDlines и усовершенствованный алгоритм EDlines в библиотеке OpenCV. Был проведен статистический анализ таких параметров, как среднее время вычислений на один кадр, количество распознанных линий и процент совпадений. Результаты представлены в табл. 2. Среднее время вычислений в предложенном алгоритме распознавания линий составило 26,2 мс на один кадр. В отличие от исходного EDlines (среднее время – 22,9 мс на кадр) на распознавание линии потребовались дополнительные 3,4 мс на кадр. При этом среднее время распознавания с помощью алгоритма LSD составляло 51,8 мс на кадр, то есть 227% и 197% от времени, затраченного при использовании LSD и исходного EDlines соответственно.

Таблица 2

База данных	LSD			EDlines			Новый алгоритм		
	Время, мс	Кол-во	Совпадения, %	Время, мс	Кол-во	Совпадения, %	Время, мс	Кол-во	Совпадения, %
МН_03	56,8	571	72,3	25,7	426	82,0	32,7	256	84,6
V1_02	52,9	487	83,1	23,1	352	81,3	27,8	194	91,2
МН_05	48,1	524	57,5	24,0	411	73,2	24,2	163	85,6
V1_03	54,5	445	62,4	21,3	348	68,7	22,9	132	79,8
V2_03	46,8	367	74,7	19,8	324	77,8	23,6	145	82,5

Как видно из табл. 2, с помощью алгоритма LSD было распознано наибольшее количество линий, на втором месте – исходный алгоритм EDlines, а результаты применения усовершенствованного алгоритма EDlines оказались самыми низкими. После того как в алгоритм EDlines был добавлен этап фильтрации длины, чтобы исключить распознавание слишком коротких линейных сегментов, количество распознанных линейных сегментов существенно сократилось, а доля более протяженных линейных сегментов среди них увеличилась. Таким образом, благодаря усовершенствованию возросла доля совпавших линейных сегментов на двух кадрах изображения.

Обозначим число совпавших линейных сегментов как m_s , а общее число распознанных линейных сегментов – как m и вычислим процент совпадений по формуле $\frac{m_s}{m} \times 100\%$. Значения m_s и m становятся известны после того, как OpenCV закончит обработку изображения. Посредством предложенного алгоритма было выделено на 63% меньше линий на кадр, чем посредством LSD, и на 52% меньше, чем за счет исходного алгоритма EDlines. Кроме того, усовершенствованный алгоритм EDlines позволил получить на 16% больше совпадений, чем LSD, и на 9,7%, чем исходный EDlines. Эффективность выделения линий с помощью усовершенствованного алгоритма EDlines по сравнению с исходным проиллюстрирована на рис. 7.

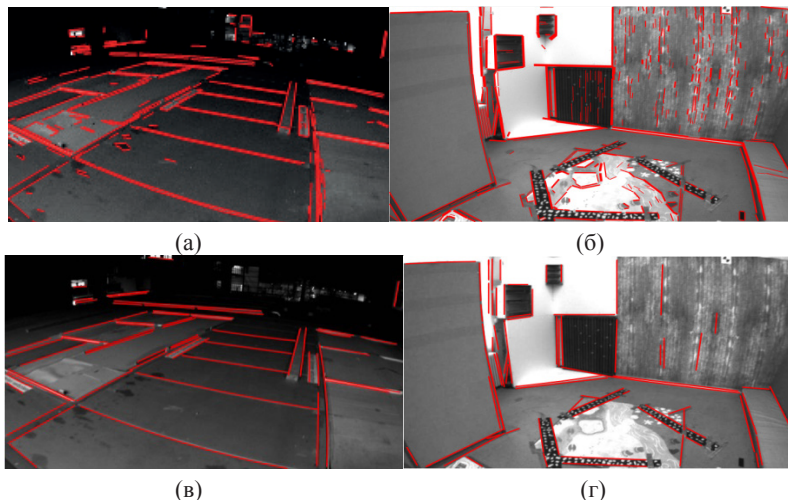


Рис. 7. Выделение линейных сегментов с помощью алгоритма EDlines (а, б); выделение линейных сегментов с помощью усовершенствованного алгоритма EDlines (в, г)

В этом эксперименте выходная частота базы данных EuRoC составляла около 20 Гц, а время, за которое были выделены ключевые точки, – около 13 мс. Общее время,

затраченное непосредственно на определение ключевых точек и линий с помощью описываемого алгоритма, составило около 28 мс, что отвечает требованиям к работе в режиме реального времени, тогда как время распознавания линий посредством LSD превысило установленные предельные значения.

Результаты показывают, что и LSD, и исходный алгоритм EDlines выделили очень много лишних линейных сегментов, в том числе слишком коротких или идентичных линий, расположенных близко друг к другу. Такие избыточные линейные сегменты могут помешать работе системы визуального SLAM в режиме реального времени и повысить вероятность ошибочных совпадений; при этом будут расходоваться значительные вычислительные ресурсы. Судя по результатам экспериментов, усовершенствованный алгоритм EDlines требует несколько больше времени на этот процесс по сравнению с исходным алгоритмом, но позволяет распознать наиболее полезные ключевые элементы, причем в режиме реального времени. Благодаря этому уменьшается вычислительная нагрузка, связанная с коррекцией ошибок при репроекции линий. Кроме того, предложенный алгоритм обеспечивает распознавание высококачественных линейных сегментов и на 11% больше совпадений, чем исходный алгоритм, что говорит о его более высокой надежности.

3.2 Проверка эффективности применения линейных сегментов

Чтобы проверить эффективность привлечения линий к процессу распознавания, необходимо сравнить погрешность траектории монокулярной SLAM-системы с другими современными системами. Из базы данных TUN RGB-D взяты последовательности реалистичных изображений внутренней обстановки помещений, в том числе с плохой контрастностью и слабой текстурой, а также обстановки с разными текстурами. В общей сложности было выбрано 13 последовательностей изображений четырех разновидностей. Целью эксперимента было сравнить предложенный алгоритм с алгоритмом PL-SLAM [7], основанным на определении ключевых точек и линий, и с алгоритмом ORB-SLAM [4], с помощью которого выделяются только ключевые точки.

В табл. 3 приведены погрешности оценивания параметров движения (в см) для каждого алгоритма, в котором использовалась база данных TUM RGB-B. Результаты применения предложенного алгоритма представлены как средние расчетные значения пяти выполненных процедур. Эффект добавления линейных сегментов при точно определенной ошибке репроекции в системах SLAM отражает сценарий fr1_floor. Очевидно, что линейный сегмент существенно улучшает эффективность работы монокулярного SLAM-метода в отсутствие особых точек в слаботекстурированных и низкоконтрастных изображениях. Кроме того, если особые точки присутствуют в достаточном количестве, чтобы поддерживать способность системы к отслеживанию, добавление линий может повысить точность позиционирования с помощью монокулярного SLAM-метода.

Результаты, представленные в табл. 3, демонстрируют превосходную устойчивость линий в чисто плоских изображениях при изменении масштаба, расстояния и при поворотах, что может значительно улучшить работу алгоритмов монокулярного SLAM, как, например, в случаях обработки изображений fr2_desk, fr3_str_tex_far, fr3_nstr_tex_far и fr3_nstr_tex_near. Эффективность определения ключевых элементов в реальном времени и достоверность выделенных линейных сегментов строго

оценивались с точки зрения отслеживания объектов в динамике. Предложенная система продемонстрировала хорошие результаты в различных сценариях, в том числе в fr3_sit_halfsph, fr3_walk_xyz и fr3_walk_halfsph. Точно такие же выводы можно сделать из табл. 4, представленной в следующем разделе.

Таблица 3

Последовательность TUM RGB-D		PL-SLAM		ORB-SLAM	Новый алгоритм	
		линия	точка + линия	точка	линия	точка + линия
Тестирование	fr1_xyz	1,46	1,21	1,38	1,29	1,08
	fr2_xyz	1,49	0,43	0,36	0,67	0,59
Построение карты	fr1_floor	9,42	7,59	8,71	5,88	3,32
	fr2_desk	4,32	-	2,36	1,94	1,75
	fr2/360_kidnap	60,11	3,92	4,99	29,39	2,43
Текстура	fr3_str_tex_far	1,25	0,89	0,98	1,14	1,09
	fr3_str_tex_near	7,47	1,25	1,55	1,86	1,43
	fr3_nstr_tex_far	37,60	-	-	18,33	4,67
	fr3_nstr_tex_near	1,58	2,06	2,88	1,66	1,41
	fr3_long_office	5,33	1,97	4,05	3,28	1,52
Объект в динамике	fr3_sit_halfsph	9,05	1,31	1,48	4,72	1,13
	fr3_walk_xyz	-	1,54	1,64	2,30	1,21
	fr3_walk_halfsph	-	1,60	2,09	-	1,36
Среднее значение		12,64	2,16	2,70	6,035	1,77

Примечание. Знак «-» показывает, что в какой-то момент отслеживание было прервано и система не обработала значительную часть последовательности изображений.

3.3 Эксперимент на открытой базе данных

Описывается визуально-инерциальный алгоритм, который предусматривает включение линий в систему VI ORB-SLAM [5]. Его эффективность оценивали по точности определения местоположения в сравнении с алгоритмом VI ORB-SLAM, основанным на распознавании ключевых точек, и с алгоритмом PL-VIO, позволяющим выявлять ключевые точки и линии и интегрировать их с данными ИИМ [10]. Экспериментальное моделирование выполнено на 11 последовательностях изображений из базы данных EuRoC. Погрешности оценивания параметров движения (в см) для каждого алгоритма, в котором используется база данных EuRoC, приведены в табл. 4. Результаты для предложенного в этой работе алгоритма представлены как средние расчетные значения пяти выполненных процедур. Для оценивания точности алгоритма привлекается абсолютная погрешность определения траектории, т.е. среднеквадратическая погрешность евклидова расстояния между расчетным и фактическим местоположением. Для согласования данных и расчета погрешности при сравнении с фактической траекторией было задействовано программное приложение EVO, чтобы оценить одометрию и SLAM. Идеальный масштабный коэффициент $|\hat{S}|$ измеряется путем приведения расчетной траектории ключевого кадра максимально близко к фактической. Погрешность масштаба можно рассчитать по формуле $\frac{|S^* - \hat{S}|}{|\hat{S}|} \times 100\%$.

Предложенный алгоритм превосходит два других на 8 из 11 последовательностей. При анализе рабочей погрешности алгоритма отмечается и более высокая точность оценивания параметров движения благодаря более эффективным ограничениям для линий в базе данных EuRoC. Наш метод существенно лучше PL-VIO с фильтрацией сегментов по длине в таких последовательностях, как MH_01_easy, MH_02_easy и MH_03_medium, которые содержат большое количество линий. В то время как ограничения для избыточных линий позволяют повысить устойчивость отслеживания, поиск лишних ключевых элементов может помешать работе алгоритма в режиме реального времени, а нелинейная оптимизация – негативно повлиять на точность всей системы в целом. В последовательностях изображений со слабой контрастностью и текстурой, например V1_02_medium, V1_03_difficult, V2_02_medium и V2_03_difficult, добавление линий повышает точность нашего метода по сравнению с VI ORB-SLAM. Интересно, что последний не позволяет отследить весь набор данных в сложном сценарии V1_03_difficult с меняющимся освещением и размытым изображением. Это еще раз доказывает положительное влияние ограничений для линий на оценивание параметров движения. Если сравнивать погрешность масштабного коэффициента у VI ORB-SLAM и у предложенного алгоритма с включением линий, погрешность меньше у первого за счет его оптимизации методом группового уравнивания точек и обнаружения замыкания циклов. Тем не менее в сценариях V1 и V2 с большим количеством участков со слабо выраженной текстурой новый алгоритм с ограничениями линий демонстрирует значительное уменьшение погрешности масштабного коэффициента по сравнению с VI ORB-SLAM, а также более высокую точность.

Таблица 4

Последовательность EuRoC	PL-VIO	VI ORB-SLAM		Новый алгоритм	
	точка + линия + ИИМ	точка + ИИМ		точка + линия + ИИМ	
	СКП, см	СКП, см	Масштабный коэффициент	СКП, см	Масштабный коэффициент
V1_01_easy	6,5	2,7	0,9	4,4	1,1
V1_02_medium	9,3	2,8	0,8	2,3	0,7
V1_03_difficult	14,1	–	–	5,6	2,1
V2_01_easy	5,8	3,2	0,2	4,1	0,7
V2_02_medium	9,4	4,1	1,4	3,5	0,8
V2_03_difficult	26,0	7,4	0,7	6,9	0,4
MH_01_easy	8,4	7,5	0,5	6,7	1,0
MH_02_easy	7,5	8,4	0,8	5,8	0,9
MH_03_medium	11,4	8,7	1,5	7,2	1,1
MH_04_difficult	12,2	21,7	3,4	16,9	2,1
MH_05_difficult	14,7	8,2	0,5	8,0	0,4
Среднее значение	11,39	7,47	1,07	6,49	1,03

Примечания. 1. Масштаб расчетной траектории ключевого кадра задан в максимальном соответствии с фактической траекторией. 2. Знак «–» показывает, что в какой-то момент отслеживание было прервано и система не обработала значительную часть последовательности изображений.

Время работы канала отслеживания для предложенного алгоритма, ORB-SLAM3 [26] и PL-SLAM, а также для улучшенного алгоритма распознавания линий EDlines представлено в табл. 5. Как видно из таблицы, предложенный алгоритм выигрывает от включения процедуры предварительной обработки данных ИИМ, в результате которой определение начального местоположения, отслеживание карты и приня-

тие решения о ключевом кадре происходят за более короткое время по сравнению с алгоритмом PL-SLAM. Кроме того, предложенный алгоритм сокращает срок распознавания линии примерно на 50% за счет использования улучшенного алгоритма EDlines. Вместе с тем этап отслеживания в предложенном алгоритме характеризуется чуть большей длительностью, чем в современном алгоритме ORB-SLAM3, зато ввод ограничений для линий приводит к некоторой экономии времени при отслеживании локальной карты. Время на отслеживание, затраченное при использовании каждого алгоритма, свидетельствует о том, что предложенный визуально-инерциальный алгоритм, сочетающий распознавание уточненных линий и ключевых точек, имеет значительное преимущество в скорости по сравнению с PL-SLAM. Несмотря на то что предложенный алгоритм немного отстает от ORB-SLAM3 при работе в режиме реального времени, его эффективность близка к оптимальной.

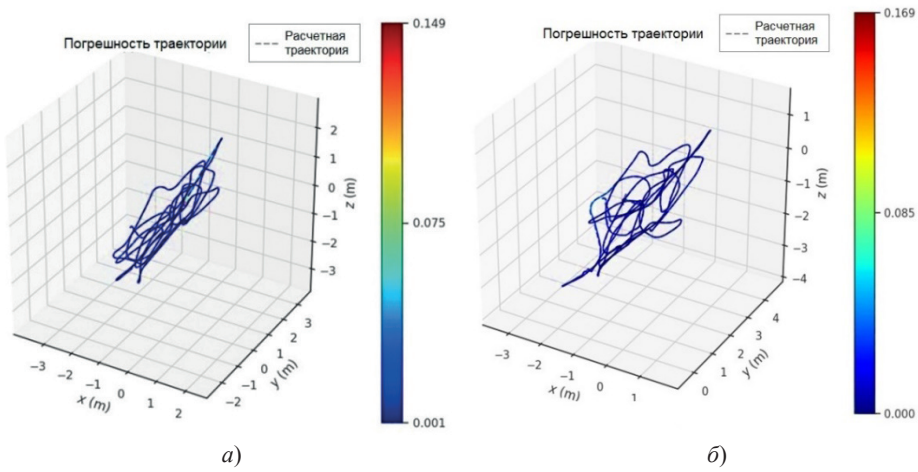


Рис. 8. Сравнение эффективности нового алгоритма (а) и VI ORB-SLAM (б) при работе с набором данных V102. Справа – контрольная цветовая шкала погрешности с увеличением абсолютной погрешности траектории снизу вверх

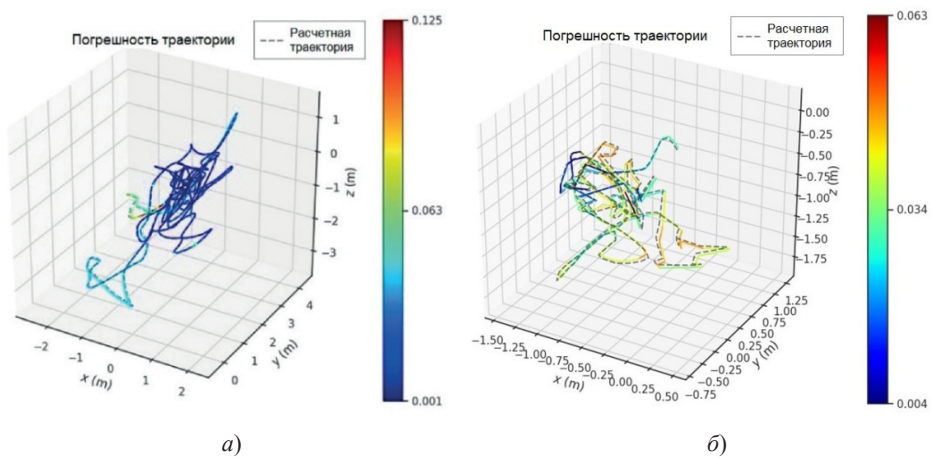


Рис. 9. Сравнение эффективности нового алгоритма (а) и VI ORB-SLAM (б) при работе с набором данных V103. Справа – контрольная цветовая шкала погрешности с увеличением абсолютной погрешности траектории снизу вверх

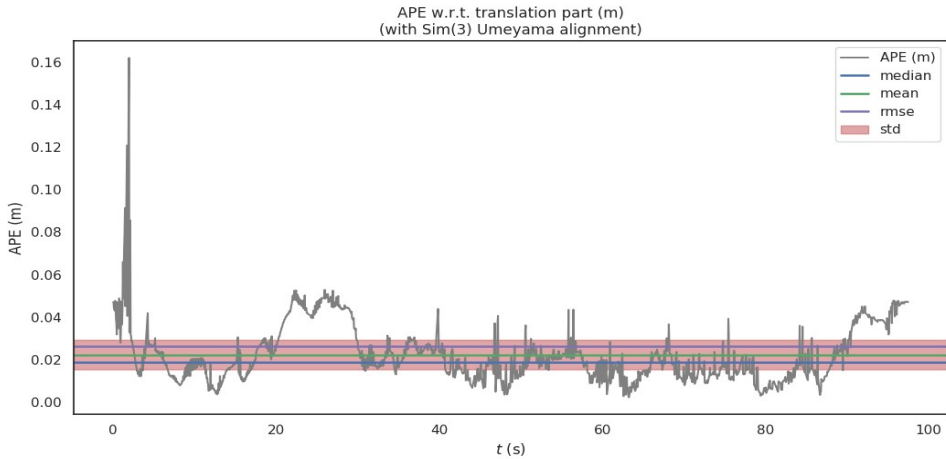


Рис. 10. Оценка работы предложенного алгоритма с набором данных V1_02

Т а б л и ц а 5

Отслеживание	ORB-SLAM3	PL-SLAM	Новый алгоритм
Распознавание ключевой точки	11,98	10,76	12,49
Распознавание линии	–	28,56	14,31
Предварительная обработка данных ИИМ	0,29	–	0,32
Оценивание начального местоположения	0,52	7,16	2,63
Отслеживание локальной карты	10,74	12,58	10,22
Выбор нового ключевого кадра	0,08	0,32	0,17
Итого, мс	23,6	59,9	40,6
Частота, Гц	20	5	10

Примечания. 1. Знак «–» показывает, что у алгоритма нет этого процесса для отслеживания. 2. Если частота циклов алгоритма составляет не менее 10 Гц, считается, что он работает в реальном времени.

4. Заключение

Для решения проблемы неудовлетворительного отслеживания ключевых точек в условиях помещений со слабо выраженной текстурой в статье был предложен алгоритм визуально-инерциальной одометрии в комбинации с определением ключевых точек и линий. Улучшенный алгоритм EDlines для распознавания линий успешно исключает из процесса идентификации линейные сегменты низкого качества и уменьшает долю ложных совпадений линий, что обеспечивает значимые и эффективные ограничения для одометрического оценивания. Благодаря учету ошибок репроекции точек и линий, погрешностей ИИМ и априорной информации в процессе нелинейной оптимизации отслеживание ключевой точки становится более устойчивым, а также повышается точность позиционирования, особенно в помещениях со слабой текстурой. Выполнена проверка предложенного улучшенного алгоритма EDlines для распознавания линий и алгоритма одометрии с использованием баз данных EuRoC и TUM RGB-D. Результаты эксперимента продемонстрировали более высокую точность позиционирования, надежность и способность обрабатывать данные в режиме реального времени по сравнению с существующими методами.

С учетом того что возможности предложенной системы ограничены для применения в крупномасштабных пространствах, в дальнейшем планируется ее оптимизировать за счет использования метода группового уравнивания точек и обнаруже-

ния замыкания циклов в системе визуально-инерциальной одометрии. Кроме того, будут исследованы перспективы развития семантического SLAM-метода и распространения описанного подхода с систем монокулярного зрения на стереовизуальные системы с дополнительным привлечением информации о плоскостях.

ЛИТЕРАТУРА

1. Liu, Y.J., Zhang, Y.Z., Rong, L., Jiang, H., and Deng, Y., Visual odometry based on the direct method and the inertial measurement unit, *Robot*, 2019, vol. 41, no. 5, pp. 683–689. <https://doi.org/10.13973/j.cnki.robot.180601>
2. Pan, L.H., Tian, F.Q., Ying, W.J., Liang, W.G., and She, B., VI-SLAM algorithm with camera-IMU extrinsic automatic calibration and online estimation, *Chinese Journal of Scientific Instrument*, 2019, vol. 40, no. 6, pp. 56–67. <https://doi.org/10.19650/j.cnki.cjsi.J1904954>
3. Mourikis, A.I., and Roumeliotis, S.I., A multi-state constraint Kalman filter for vision-aided inertial navigation, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3565–3572. <https://doi.org/10.1109/ROBOT.2007.364024>
4. Mur-Artal, R., Montiel, J.M.M., and Tardós, J.D., ORB-SLAM: A versatile and accurate monocular SLAM system, *IEEE Transactions on Robotics*, 2015, vol. 31, no. 5, pp. 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>
5. Mur-Artal, R., and Tardós, J.D., Visual-inertial monocular SLAM with map reuse, *IEEE Robotics and Automation Letters*, 2017, vol. 2, no. 2, pp. 796–803. <https://doi.org/10.1109/LRA.2017.2653359>
6. Qin, T., Li, P., and Shen, S., VINS-Mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Transactions on Robotics*, 2018, vol. 34, no. 4, pp. 1004–1020. <https://doi.org/10.1109/TRO.2018.2853729>
7. Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., and Moreno-Noguer, F., PL-SLAM: Real-time monocular visual SLAM with points and lines, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4503–4508. <https://doi.org/10.1109/ICRA.2017.7989522>
8. Gioi, R., Jakubowicz, J., Morel, J.-M., and Randall, G., LSD: A fast line segment detector with a false detection control, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, vol. 32, no. 4, pp. 722–732. <https://doi.org/10.1109/TPAMI.2008.300>
9. Gomez-Ojeda, R., Briales, J., and Gonzalez-Jimenez, J., PL-SVO: Semi-direct monocular visual odometry by combining points and line segments, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4211–4216. <https://doi.org/10.1109/IROS.2016.7759620>
10. He, Y.J., Zhao, J., Gao, Y., He, W., and Yuan, K., PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features, *Sensors*, 2018, vol. 18, no. 4, pp. 1159–1184. <https://doi.org/10.3390/s18041159>
11. Fu, Q., Wang, J.L., Yu, H.H., Islam, A., Guo, F., and Zhang, H., PL-VINS: Real-time monocular visual-inertial SLAM with point and line, *arXiv*, 2020, preprint arXiv:2009.07462. <https://doi.org/10.48550/arXiv.2009.07462>
12. Shu, F.W., Wang, J.X., and Pagani, A., Structure PLP-SLAM: Efficient sparse mapping and localization using point, line, and plane for monocular, RGB-D and stereo cameras, *ArXiv*, 2022, preprint ArXiv 2207 06058, <https://doi.org/10.48550/arXiv.2207.06058>
13. Yoon, S., and Kim, A., Line as a visual sentence: Context-aware line descriptor for visual localization, *IEEE Robotics and Automation Letters*, 2021, vol. 6, no. 4, pp. 8726–8733. <https://doi.org/10.1109/LRA.2021.3111760>
14. Yunus, R., Li, Y., and Tombari, F., Manhattan SLAM: Robust planar tracking and mapping leveraging mixture of Manhattan frames, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6687–6693. <https://doi.org/10.1109/ICRA48506.2021.9562030>
15. Wu, T.-H., and Chen, K.-W., LGC Net: Feature enhancement and consistency learning based on local and global coherence network for correspondence selection, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 6182–6188. <https://doi.org/10.1109/ICRA48891.2023.10160290>
16. Akinlar, C., and Topal, C., Edlines: Real-time line segment detection by edge drawing, *Proc. 18th IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 2837–2840. <https://doi.org/10.1016/j.patrec.2011.06.001>
17. Lupton, T., and Sukkarieh, S., Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions, *IEEE Transactions on Robotics*, 2012, vol. 28, no. 1, pp. 61–76. <https://doi.org/10.1109/TRO.2011.2170332>

18. Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D., On-manifold pre-integration for real-time visual-inertial odometry, *IEEE Transactions on Robotics*, 2017, vol. 33, no. 1, pp. 1–21. <https://doi.org/10.1109/TRO.2016.2597321>
19. Mur-Artal, R., and Tardós, J.D., ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras, *IEEE Transactions on Robotics*, 2017, vol. 33, no. 5, pp. 1255–1262. <https://doi.org/10.1109/TRO.2017.2705103>.
20. Kschischang, F.R., Frey, B.J., and Loeliger, H.-A., Factor graphs and the sum-product algorithm, *IEEE Transactions on Information Theory*, 2001, vol. 47, no. 2, pp. 498–519. <https://doi.org/10.1109/18.910572>.
21. Dellaert, F., and Kaess, M., Square root SAM: Simultaneous localization and mapping via square root information smoothing, *International Journal of Robotics Research*, 2006, vol. 25, no. 12, pp. 1181–1203. <https://doi.org/10.1177/0278364906072768>
22. Kümmerle, R., Grisetti, G., Strasdat, H.M., Konolige, K., and Burgard, W., G2o: A general framework for graph optimization, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613. <http://doi.org/10.1109%2FICRA.2011.5979949>
23. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P., Keyframe-based visual-inertial odometry using nonlinear optimization, *International Journal of Robotics Research*, 2015, vol. 34, no. 3, pp. 314–334. <https://doi.org/10.1177/0278364914554813>
24. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., and Siegwart, R., The EuRoC micro aerial vehicle datasets, *International Journal of Robotics Research*, 2016, vol. 40, no. 6, pp. 1157–1163. <https://doi.org/10.1177/0278364915620033>
25. Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D., A benchmark for the evaluation of RGB-D SLAM systems, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580. <https://doi.org/10.1109/IROS.2012.6385773>
26. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., and Tardós, J.D., ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM, *IEEE Transactions on Robotics*, 2021, vol. 37, no. 6, pp. 1874–1890. <https://doi.org/10.1109/TRO.2021.3075644>

Yang Gang, Meng WeiDa, Hou GuoDong, Feng NingNing (Xi'an University of Posts and Telecommunications, Xi'an, Republic of China).

Real-time Visual-Inertial Odometry based on Point-Line Feature Fusion, *Giroskopiya i Navigatsiya*, 2023, vol. 31, no. 4 (123), pp. 96–117.

Abstract. *To improve the localization accuracy and tracking robustness of monocular feature-based visual SLAM systems in low-texture environments, a visual-inertial odometry method combining line features and point features is proposed, taking advantage of the easy availability of line features in real-world environments and the high accuracy of feature-based methods. The combination of point and line features ensures accurate positioning of the SLAM system in low-texture environments, while the inclusion of IMU data provides prior information and scale information. The pose is optimized by minimizing the reprojection error of point and line features and the IMU error using bundle adjustment. An improved EDlines algorithm is introduced, which incorporates a pixel chain length suppression process to enhance the effectiveness of extracted line features and reduce the rate of line feature misalignment. Experimental results on the public EuRoC dataset and TUM RGB-D dataset show that the proposed method meets the real-time requirements and has higher localization accuracy and robustness compared with the visual SLAM method based on single point feature or the method adding traditional line features.*

Key words: simultaneous localization and mapping, visual-inertial odometry, point-line features, nonlinear optimization.

Материал поступил 19.07.2023