



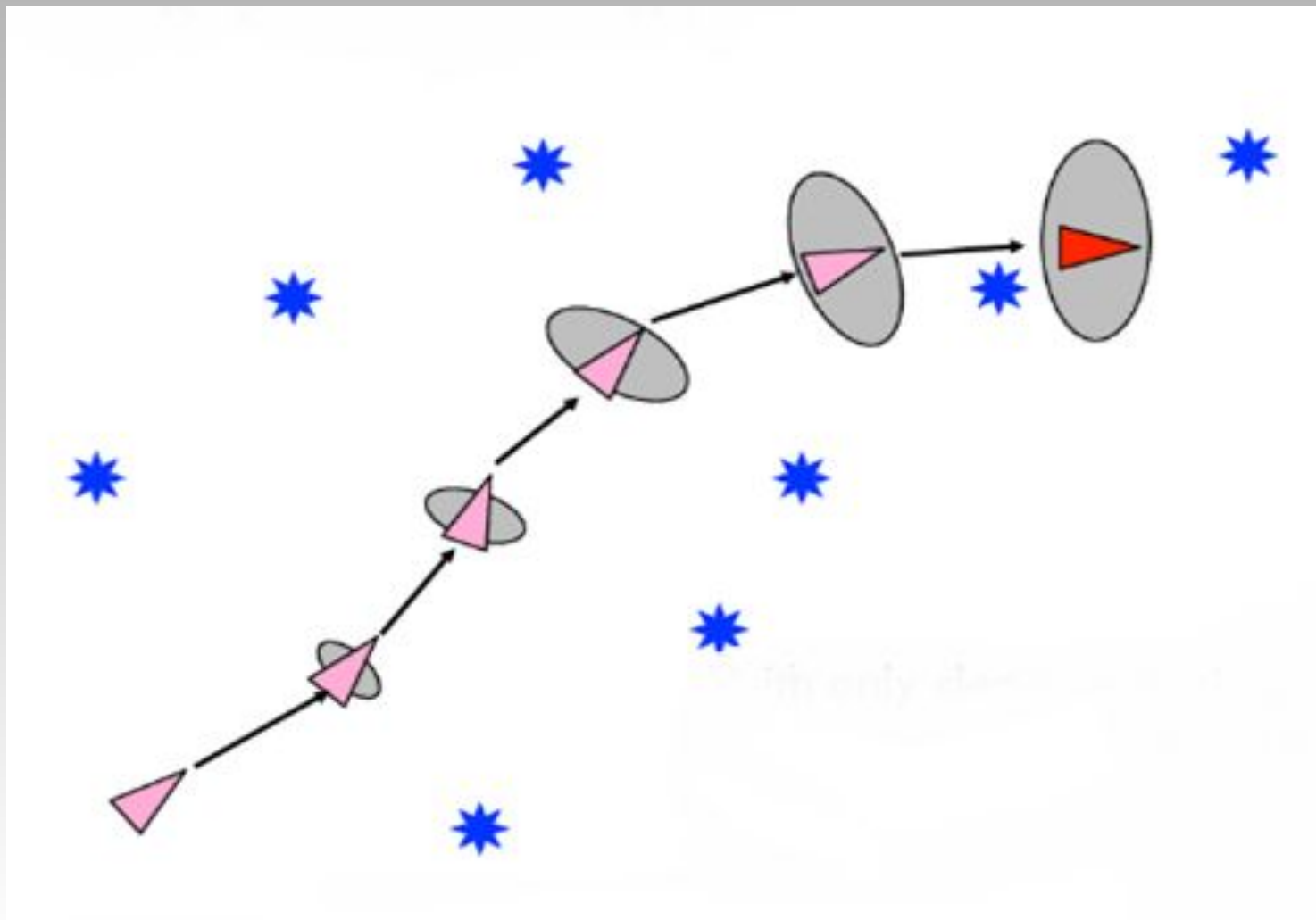
**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ПЕТРА ВЕЛИКОГО**

Кафедра «Мехатроники и роботостроения» (При ЦНИИ РТК)

Визуальные алгоритмы SLAM

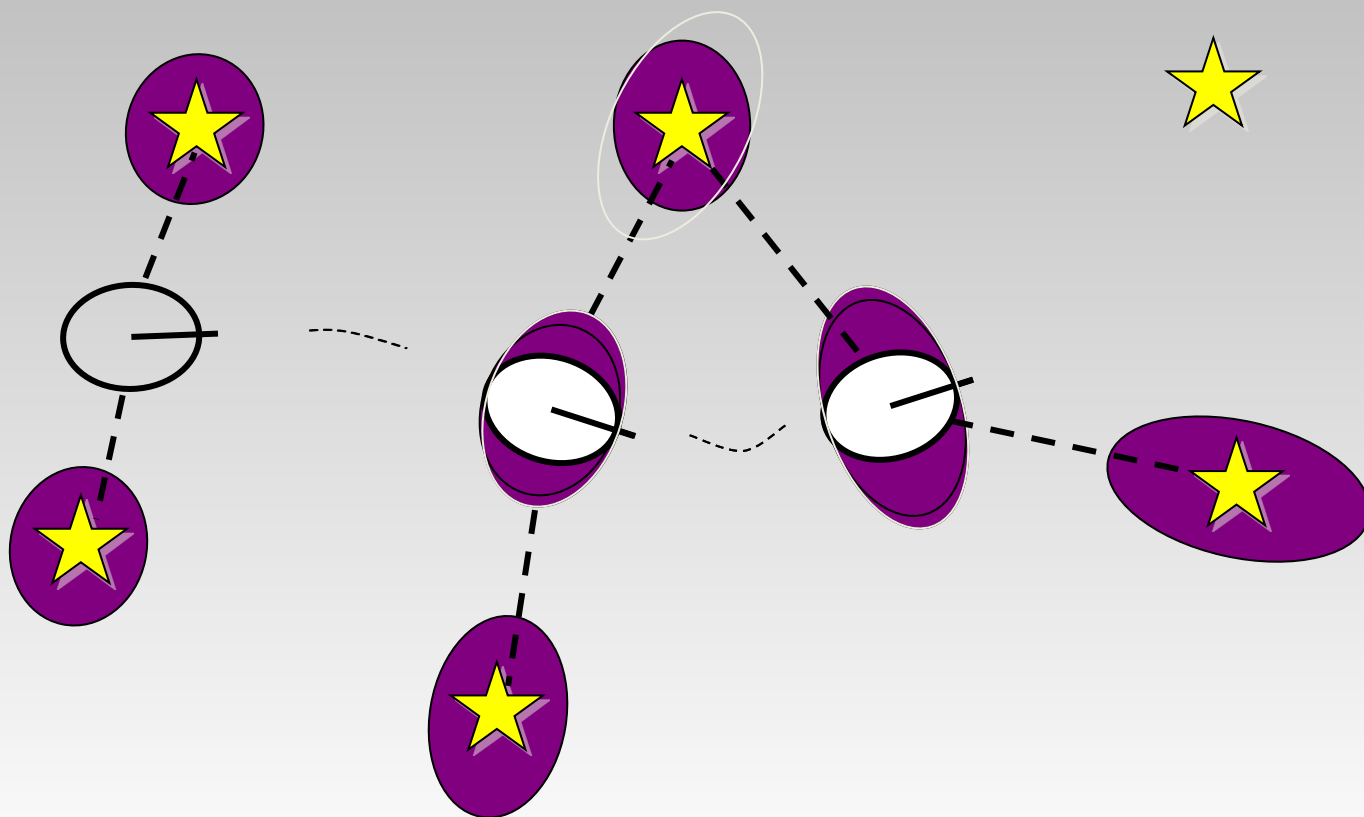
Докладывает студент СПбПУ:
И. С. Балташов

О задаче SLAM



SLAM

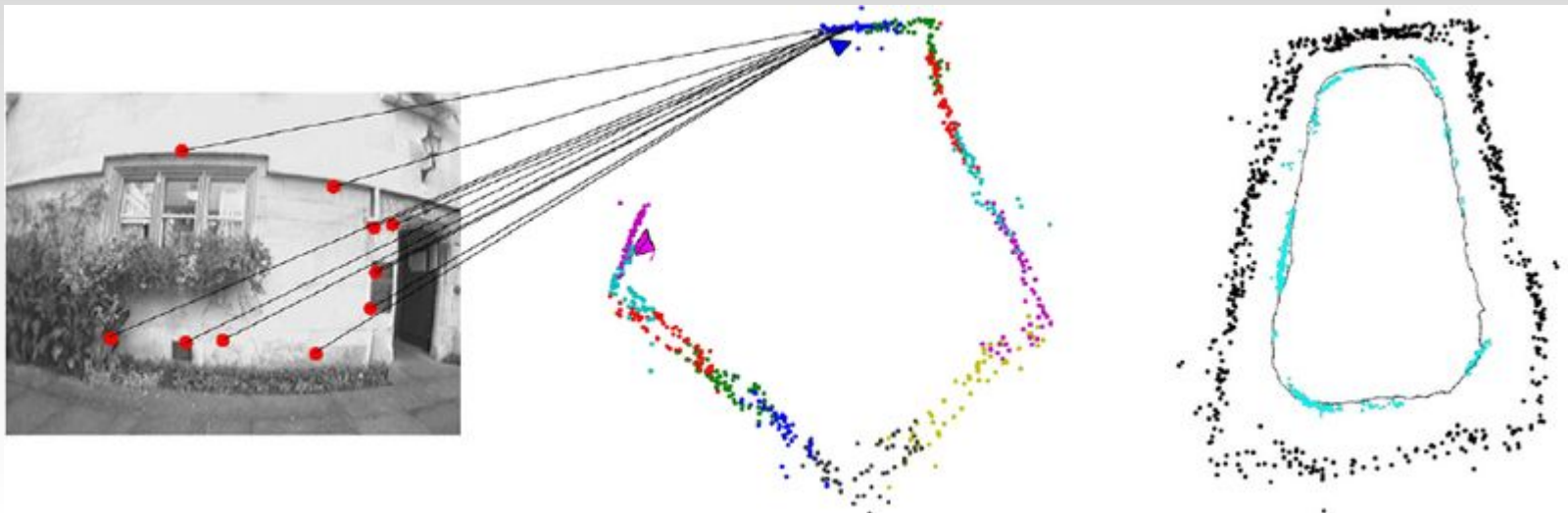
SLAM (*Simultaneous Localization and Mapping*) – методы одновременной локализации и построения карты.



В чём же разница SLAM и Визуальной одометрии

Конечно можно сказать, что в наличие карты, но есть алгоритмы визуальной одометрии которые строят карту (например, DSO или viso2).

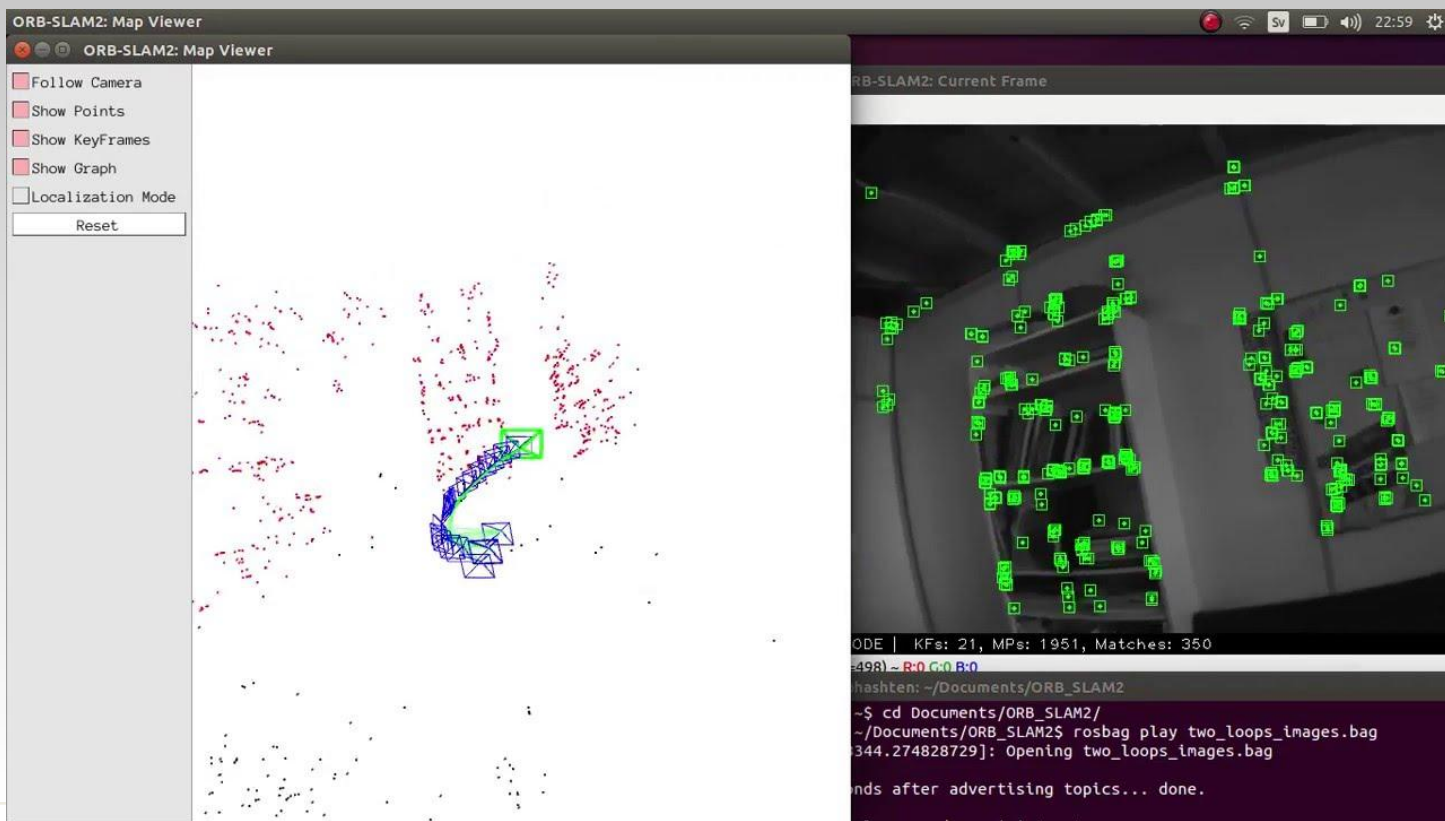
Поэтому я разделяю, по наличию loop closure detection. Loop closure – обнаружение/замыкание петель. Отвечает за обработку ситуаций, когда робот возвращается туда, где уже побывал и исправления карты с помощью новой информации.



О плотности карт

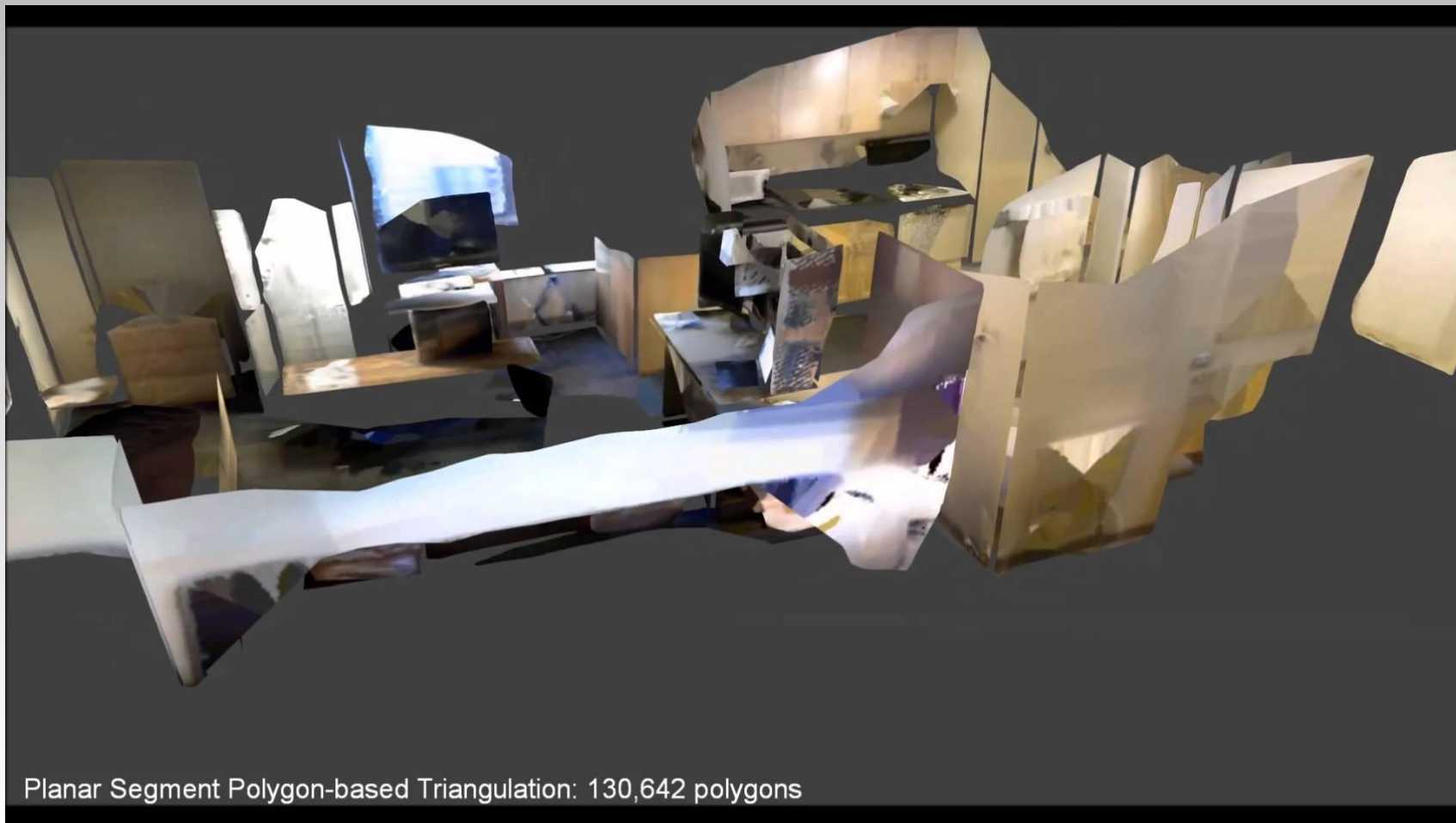
Почти все визуальные алгоритмы SLAM используют точечную карты.

Неплотная точечная (Sparse) карта – на карту добавляются лишь некоторые (ключевые) точки.



О плотности карт

Плотная точечная (Dense) карта – точки добавляются по максимуму. Есть ещё полуплотные (Semi Dense).



Прямой/косвенный метод

Косвенные методы (Indirect) – Проходят в два этапа. Во-первых, происходит предварительная обработка данных для генерации промежуточных значений. При этом решается часть общей задачи, такая как установления соответствий. Во-вторых, вычисленные промежуточные значения интерпретируются как шумные измерения в вероятностной модели для оценки геометрии и движения камеры. Чаще всего в качестве первого шага в таких методах выступает извлечение ключевых точек (feature-based), но не обязательно.

Прямые (direct) – Пропускаем шаг предварительной обработки и используем фактические значения датчика - свет, полученный от определенного направления в течение определенного периода времени. Обычно таким методом компенсируется фотометрическая ошибка.

Прямой/косвенный метод

 Comparison	
Feature-Based	Direct
can only use & reconstruct corners	can use & reconstruct whole image
faster	slower (but good for parallelism)
flexible: outliers can be removed retroactively.	inflexible: difficult to remove outliers retroactively.
robust to inconsistencies in the model/system (rolling shutter).	not robust to inconsistencies in the model/system (rolling shutter).
decisions (KP detection) based on less complete information.	decision (linearization point) based on more complete information.
no need for good initialization.	needs good initialization.
~20+ years of intensive research	~4 years of research (+5years 25 years ago)

Комбинации подходов

Sparse + Indirect: (Не плотные + не прямые) – Используем ключевые точки и их же добавляем на карты - monoSLAM, PTAM, и ORB-SLAM.

Dense + Indirect: (плотные + не прямые) – используется только с RGBD камерами или стереопарами. – RTAB-Map, RGBD-SLAM.

Dense + Direct: (плотные + прямые) – Используют фотометрическую ошибку и геометрию предшествующих позиций для нахождения плотной или полу-плотной карты – DTAM, LSD-SLAM.

Sparse + Direct: (не плотная + прямая) – Используем фотометрическую ошибку, без учёта предыдущей геометрии - DSO

Разделение по камерам

Mono-SLAM

Одна камера

Наличие дрейфа масштаба



Stereo-SLAM

Стереопара или RGBD-камера

Ограничение глубины на
который предназначены
датчики



RTAB-Map

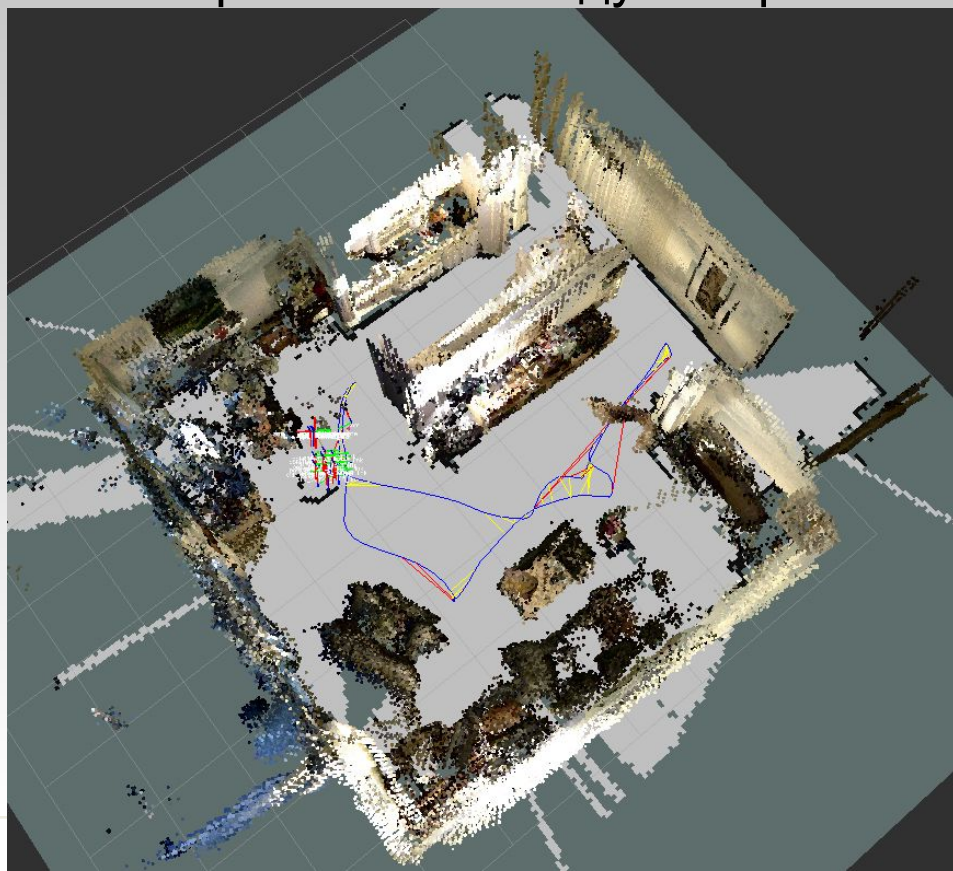
RTAB-Map (Real-Time Appearance-Based Mapping) – визуальный алгоритм SLAM на основе графа для RGB-D камер.

Использует визуальный алгоритм замыкания циклов, использующего bag-of-words (словарь слов) для оценки где находится новый ключевой кадр – прежнее или новое положение. После нахождения цикла позиции графа оптимизируются с целью минерализации ошибки в графе.

Принцип работы RTAB-Map

Каждый узел графа (ключевой кадр) содержит свою позицию в 3d пространстве, цветную 3d карту глубины и список ключевых точек карты, которые обнаружены на данном изображении. Рёбра графа отражают связь между этими узлами, связь создаётся только между узлами, между которыми переместился робот или между которыми обнаружено замыкание цикла.

В bag of words хранятся дескрипторы ключевых точек и номера узлов содержащие их.





Принцип работы RTAB-Map

- 1) Первый кадр становится ключевым. Его карта глубины добавляется на 3D кадр. А дескрипторы ключевых точек в bag of words.
- 2) Следующий ключевой кадр будет добавлен только если робот переместится на некоторое расстояние или прошёл заданный промежуток времени.
- 3) Положение нового ключевого кадра уточняется благодаря соотношению ключевых точек двух кадров (напомню, что метрика благодаря карте глубины у нас есть).
- 4) Loop close detection. Поиск соотношения между дескрипторами текущего ключевого кадра и дескрипторами в bag of words. Если количество “общих” точек у текущего ключевого кадра и у кадра с наибольшим соответствием превышает определённый порог, то происходит замыкание цикла. При этом позиция текущего кадра уточняется для совпадения ключевых кадров с предыдущим, а остальные узлы петли оптимизируется. И добавляется соответствующее новое ребро.
- 5) 3d карта нового ключевого кадра добавляется на карту. А дескрипторы ключевых кадров в bag of words. И процесс начинается с начала.

Фишки RTAB-Map

1) Большое количество дескрипторов и детекторов точек:

0=SURF 1=SIFT 2=ORB 3=FAST/FREAK 4=FAST/BRIEF
5=GFTT/FREAK 6=GFTT/BRIEF 7=BRISK 8=GFTT/ORB.

HO! If OpenCV is built **with the nonfree** module, RTAB-Map is free for research only because it depends on **SURF** and **SIFT** features. **SIFT** and **SURF** are not free for commercial use.

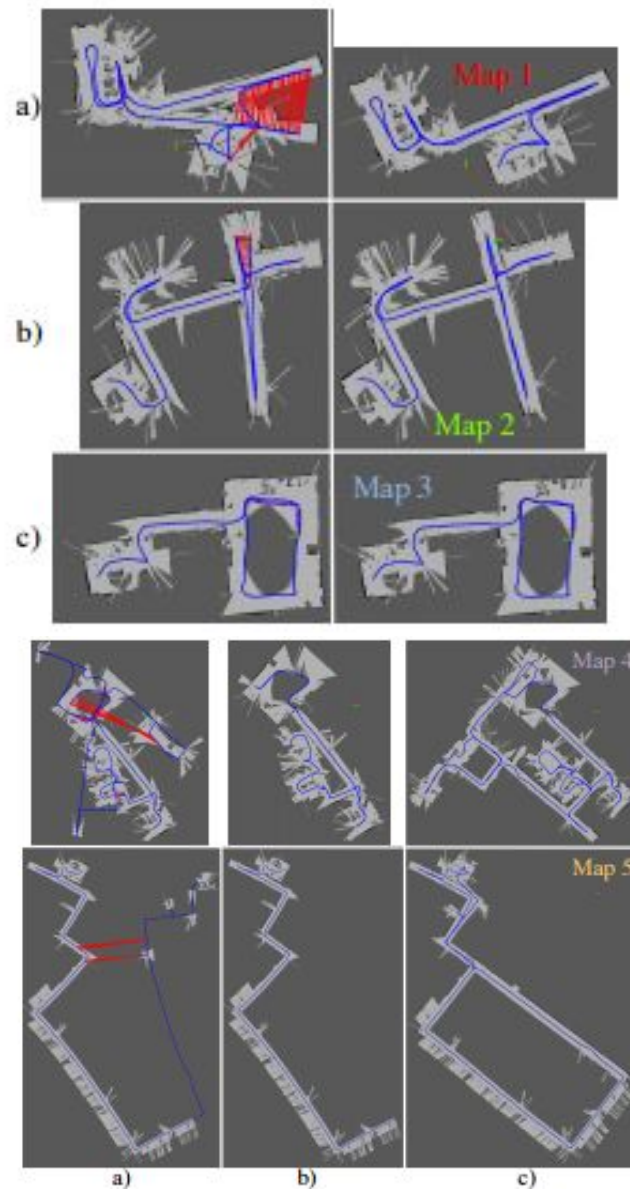
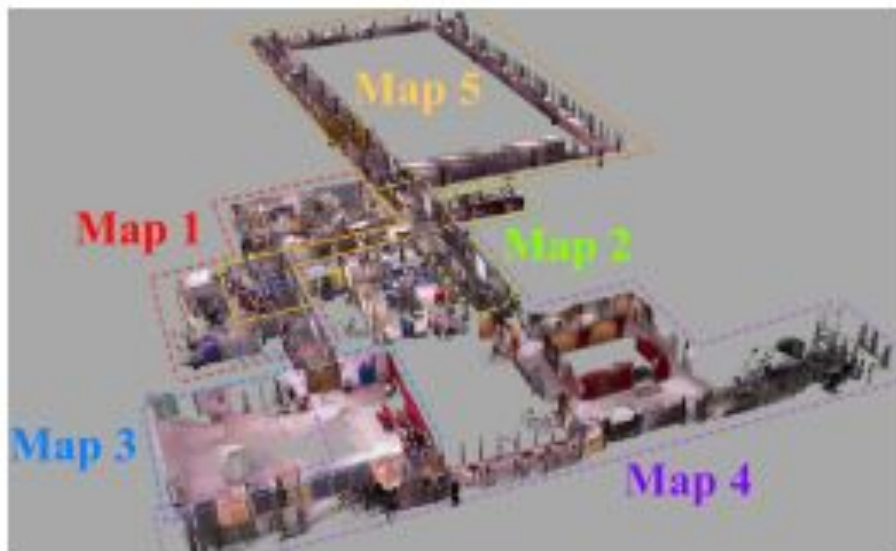
2) Возможность использования одометрии

3) Уточнение позиции благодаря построения 2d сеточной карты с помощью дальномера (ICP алгоритм).

4) Несколько вариантов оптимизации графа TORO, g2o, GTSAM.

5) Уменьшение нагрузки на память за счёт разбиения карты на карты меньшего размера.

Фишки RTAB-Map





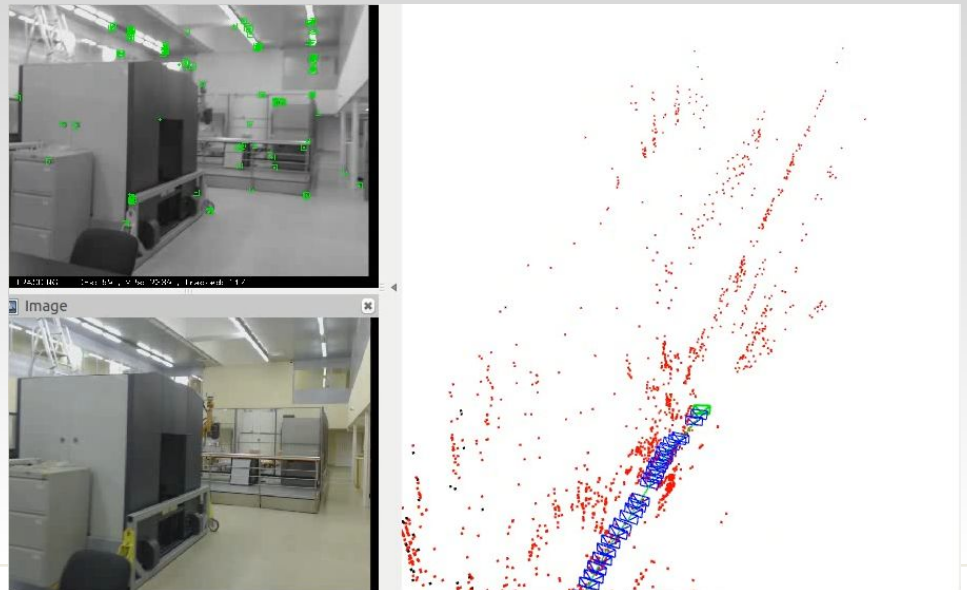
Пример работы RTAB-Map

Не забудь скачать видосик с рабочего компа

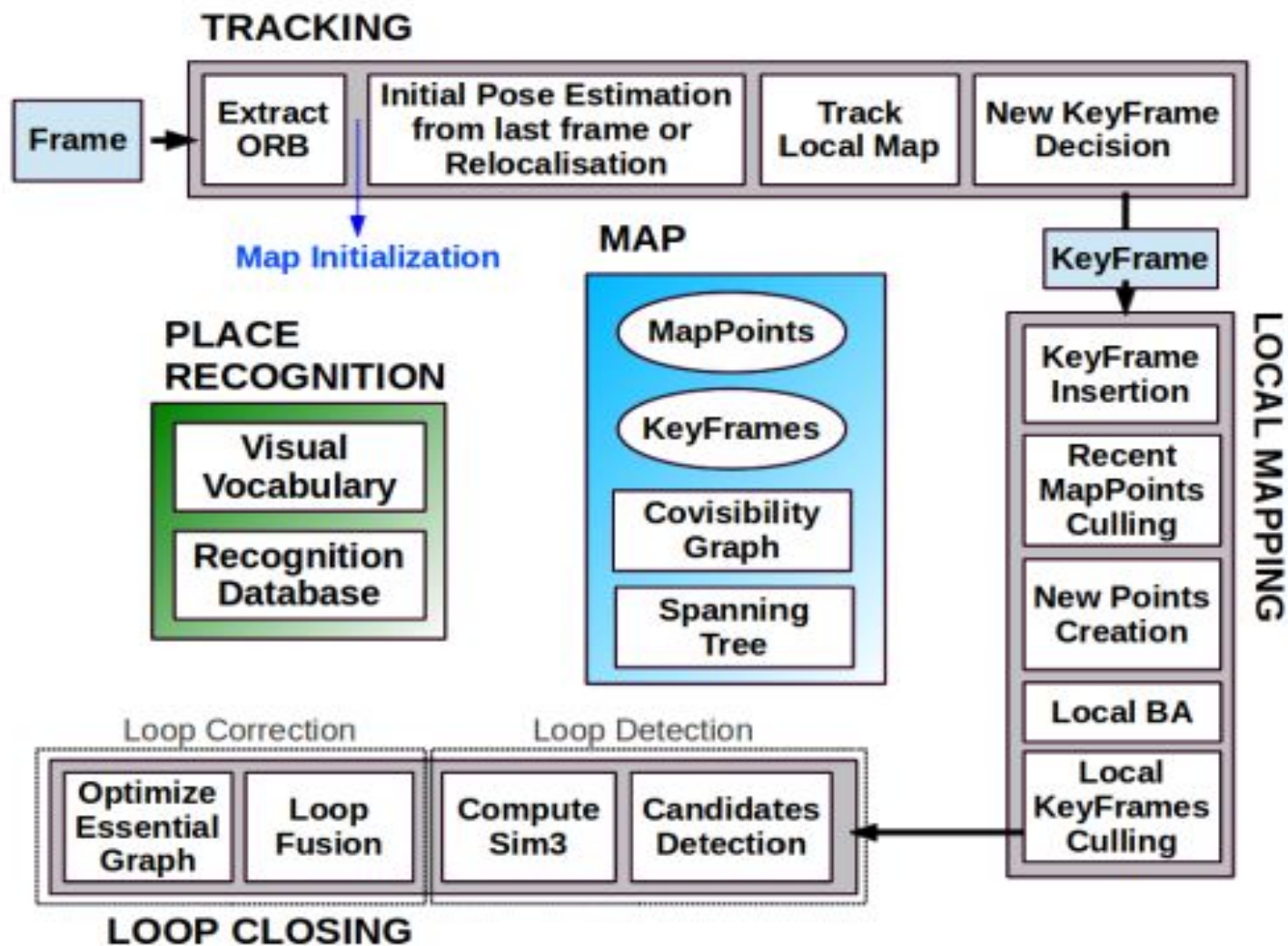
ORB-SLAM

ORB-SLAM (Oriented FAST and Rotated BRIEF SLAM)— визуальный алгоритм SLAM для моно-камер использующая детектор и дескриптор ORB, covisibility Graph и bag-of-words оптимизацию. Метод основан на отслеживании ключевых точек, в результате чего, в качестве ключевого кадра выступает изображение с набором ключевых точек вычисленных по методу FAST и найденных на обоих кадрах. В качестве дескриптора ключевых кадров используется метод BRIEF (Binary Robust Independent Elementary Features).

Так же есть модернизация данного алгоритма для применения с RGB-D камерами и стереопарами.



Принцип работы ORB-SLAM



Принцип работы ORB-SLAM

Каждая точка карты p_i состоит из:

- 3D позиция $X_{w,i}$ в глобальной системе координат
- Направления взгляда n_i , которое является средним единичным вектором всех его направлений наблюдения (лучи которые соединяют точки с оптическим центром ключевых кадров, содержащим их.)
- Дескриптор ORB D_i с минимальной функцией Хамминга среди всех кадров на которых наблюдалась эта точка
- Максимальное d_{max} и d_{min} минимальное расстояние при котором может наблюдаться точка в соответствие с пределами шкалы инвариантности функции ORB.

Ключевой кадр содержит:

- Позицию камеры T_{iw} , которое представляет собой жесткое тело преобразование, которое преобразует точки из мировой системы координат в систему координат камеры
- Параметры камеры, включая фокусное расстояние и принципиальную точку
- Все фичи извлечённые ORB из кадра ассоциированные или нет с точками карты, координаты которых не искажены или искажение предусмотрено.

Covisibility Graph (граф взаимной видимости) - Каждый узел представляет собой ключевой кадр и ребро между двумя ключевыми кадрами существует, если они наблюдают общие точки карты (по крайней мере, 15), вес ребра θ это число общих точек карты.

Essential Graph (существенный граф) - охватывающие дерево, подмножество рёбер графа взаимной видимости с большой степенью взаимной видимости ($\theta_{\min} = 100$). Нужен только для замыкания цикла.

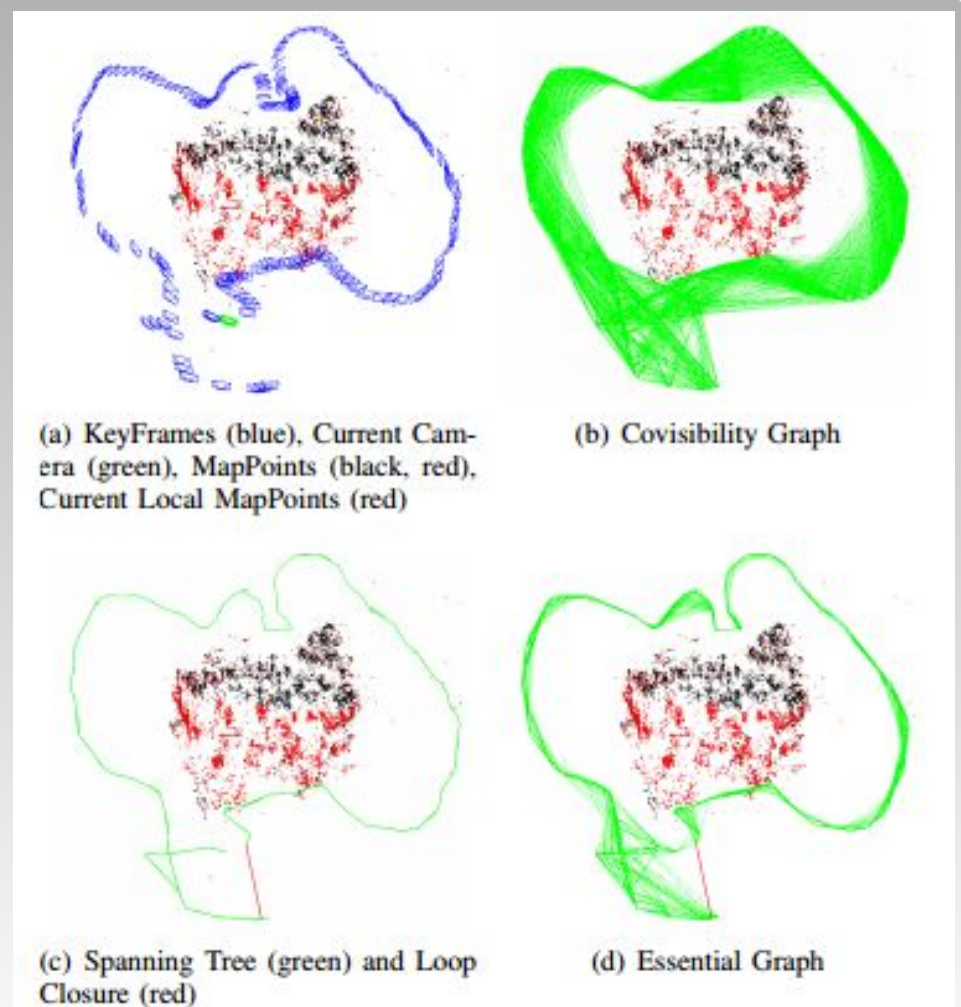


Fig. 2. Reconstruction and graphs in the sequence *fr3_long_office_household* from the TUM RGB-D Benchmark [38].

Шаг 0: Инициализация

1.1 Автоматическая инициализация карты

Целью инициализации карты является вычисление относительную позицию между двумя кадрами для триангуляции начального набора точек карты.

(1) Найти начальное соответствие: извлечённых ORB фич в текущем кадре F_c и поиск совпадений $x_c \leftrightarrow x_r$ в кадре F_r . Если найдено недостаточно значений, то поиск с новыми кадрами.

(2) Для инициализации предлагается вычислять параллельно:

- Проективное преобразование H_{cr} , если сцена плоская
- Фундаментальная матрица F_{cr} для неплоской сцены

$$x_c = H_{cr} x_r \quad x_c^T F_{cr} x_r = 0 \quad (1)$$

Шаг 0: Инициализация

Ищем с помощью RANSAC по 8 точкам для фундаментальной матрицы и по 4 для проективного преобразования. Если найти решения не получилось, то с начала. Для каждого подхода, на каждой итерации вычисляется балл.

$$S_M = \sum_i (\rho_M(d_{cr}^2(\mathbf{x}_c^i, \mathbf{x}_r^i, M)) + \rho_M(d_{rc}^2(\mathbf{x}_c^i, \mathbf{x}_r^i, M)))$$
$$\rho_M(d^2) = \begin{cases} \Gamma - d^2 & \text{if } d^2 < T_M \\ 0 & \text{if } d^2 \geq T_M \end{cases} \quad (2)$$

(3) Выбор модели по эмпирической функции

$$R_H = \frac{S_H}{S_H + S_F}$$

$R_H > 0,45$ то проективное преобразование, иначе фундаментальная матрица.

Шаг 0: Инициализация

(4) На основе полученной модели вычисляем гипотезы движения. Для проективного преобразование триангулируем все 8 гипотез и выбираем лучшее.

В случае фундаментальной матрицы, мы преобразовать его к существенной матрицы с использованием калибровочной матрицы K :

$$E_{gc} = K^T F_{gc} K$$

С помощью метода сингулярного разложения значения получаем 4 гипотезы движения. Триангулируем четыре решения и выбираем лучшее.

(5) Bundle adjustment для уточнения первоначальной реконструкции.

Bundle adjustment, чтобы не спрашивали А ещё есть страшное приложение

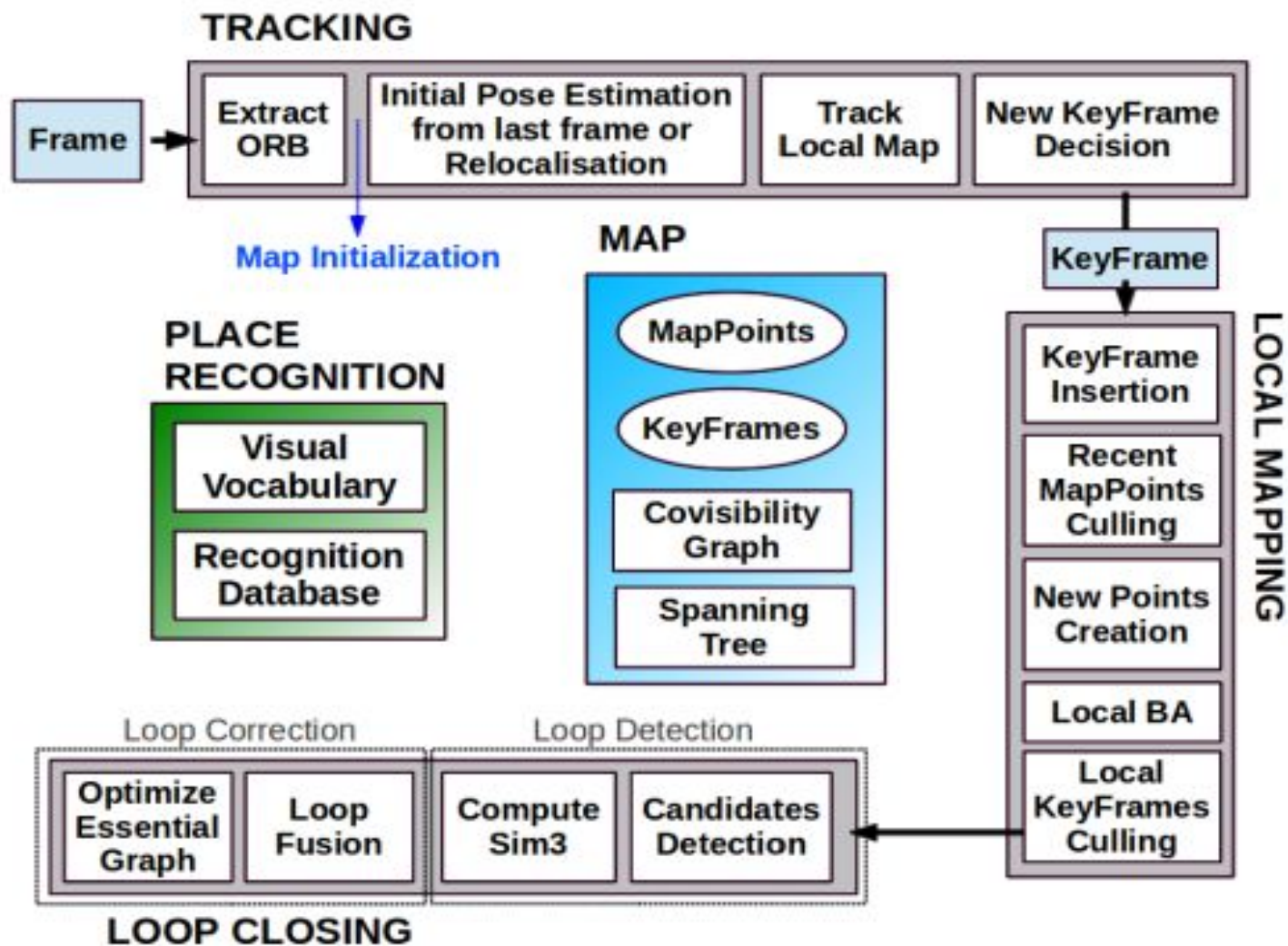
Уравнение наблюдения (связь между трёхмерными координатами $(X, Y, Z)^T$ точки и её положением на изображении $(u, v)^T$) можно записать так:

$$\begin{pmatrix} u \\ v \end{pmatrix} = k \left\{ (R) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \right\} \equiv M \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + kt$$

Получается, что нам нужно подобрать внешние (и внутренние) параметры всех камер и трёхмерные координаты точек так, чтобы это соотношение выполнялось как можно точнее для всех точек.

Можно решить эту нелинейную систему скопом нелинейными методами (bundle adjustment). Основная проблема тут очевидна — необходимо уметь решать очень большие системы нелинейных уравнений (ну а точнее — минимизировать нелинейную функцию в многомерном пространстве) и добиваться их сходимости к корректному решению

Принцип работы ORB-SLAM



Шаг 1: TRACKING

- (1) Извлечение FAST corners фич из кадра на 8 масштабных уровнях с масштабным коэффициентом 1,2. Кадр разделяется сеткой и из ячейки пытаемся извлечь хотя бы 5 фич, адаптируя порог детектора. Ищем дескрипторы ORB для фич.
- (2) Первоначальная оценка позиции.
- (3) Если трекинг был потерян, то используется глобальная релокализация.
- (4) Track Local Map

Чтобы упростить добавления позиции камеры и набора фич на карту, используется локальная карта. Эта локальная карта содержит набор ключевых кадров $K1$, которые имеют общие точки с текущим кадром, а также множество ключевых кадров $K2$ с соседями $K1$ ключевыми кадрами в covisibility графике. Локальная карта также имеет опорный ключевой кадр $K_{REF} \in K1$, который разделяет большинство точек карты с текущим кадром (общие точки).

Шаг 1: TRACKING

Теперь для каждой точки на карте из $K1$ и $K2$ производится поиск в текущем кадре соответствующий следующим образом:

- (1) Вычислить позицию точки карты на x в текущем кадре. Отбросить её, если она лежит вне границы изображения.
- (2) Вычислить угол текущего луча видимости v и среднее направление взгляда на точку n , отбросить, если $v \cdot n < \cos(60^\circ)$
- (3) Вычислить расстояние D от точки на карте до центра камеры. Отбросить, если он находится вне масштабной инвариантности области точки отображения $d / \in [D_{\min}, D_{\max}]$.
- (4) Подсчитать масштаб в кадре отношением $d/D_{\min}$.
- (5) Сравнение дескрипторов D точек карты с ещё несогласованными ORB фичами во фрейми в предсказанном масштабе, а также вблизи x и связать точку на карте с лучшим результатом.

Позиция камеры окончательно оптимизированная со всеми точками карте в кадре.

Шаг 1: TRACKING

5) Выбор нового ключевого кадра

Новый ключевой кадр может быть добавлен, если соответствует следующим условиям:

(1) Более 20 кадров должно пройти от последней глобальной релокализации.

(2) Track Local Map находится в режиме ожидания или прошло более 20 кадров с создания последнего ключевого кадра.

(3) Текущий кадр отслеживает по меньшей мере, 50 точек.

(4) Текущий кадр отслеживает менее 90% точек, чем K_{REF} .

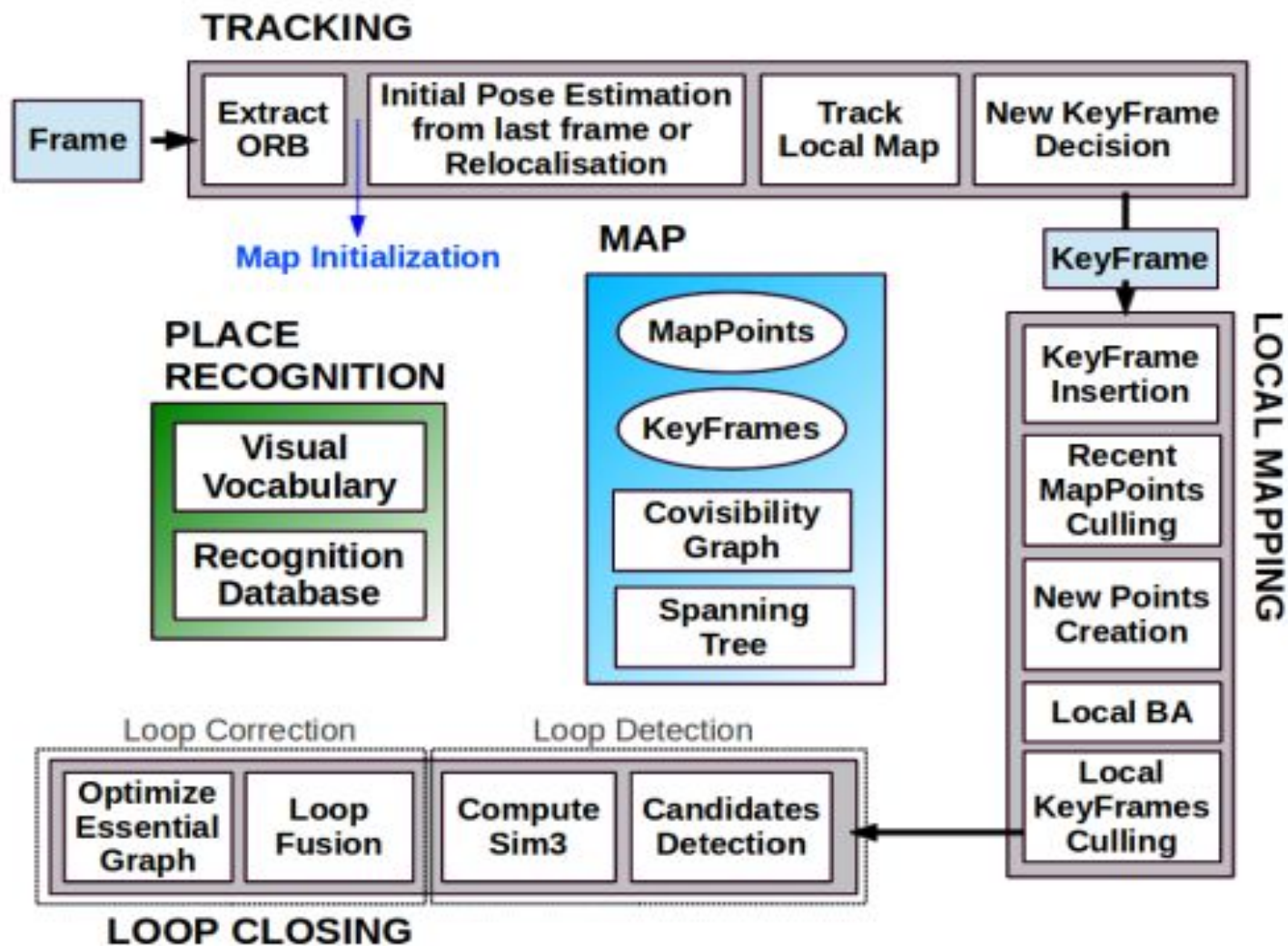
Шаг 2: LOCAL MAPPING

- 1) Вставка ключевого кадра – Обновляем covisibility граф и spanning tree. Затем мы вычисляем bags of words представления ключевых кадров, которые помогут в ассоциации данных для триангуляции новых точек.
- 2) Recent Map Points Culling – (избавимся от паразитных данных) точка будет добавлена на глобальную карту, если:
 - (1) трекинг должен находить точку более чем в 25% кадров в которых он предсказывает, что данная точка будет видимый.
 - (2) После создания точка должна наблюдаться хотя бы трёх ключевых кадров.
- 3) Создание новых точек карты.

Новые точки создаются триангуляцией фич ORB с подключённых ключевых кадров K_s в covisibility графе. Ищем совпадение фич ORB в двух кадрах. Отбрасываем те фичи, которые не выполняют эпиполярное ограничение.

Первоначально точка на карте наблюдается от двух ключевых кадров, но может быть согласована и с другими.
- 4) Local Bundle Adjustment

Принцип работы ORB-SLAM



Шаг 3: LOOP CLOSING

1) Детектор кандидатов для закрытия цикла

Сначала мы вычислим сходство между bag of words вектора K_i и всех своих соседей по covisibility графа ($\theta_{\min} = 30$) и сохранить самый низкий балл $S_{\min}$. Затем мы делаем запрос к базе данных распознавания и отбросить все эти ключевые кадры, чей балл ниже, чем $S_{\min}$. Для того, чтобы принять кандидата Loop closing мы должны обнаружить последовательно трех loop кандидатов, которые являются последовательными (ключевые кадры, связанные в covisibility графике).

2) Подсчёт подобия преобразования ($\text{sim}(3)$)

В монокулярном SLAM есть семь степеней свободы, в котором карта может плыть - три перемещения, три вращения и масштабный коэффициент. Поэтому, чтобы закрыть петлю нам необходимо вычислить преобразование подобия из текущего ключевого кадра K_i в ключевом кадре петли K_j , который сообщает нам об ошибке, накопленной в цикле. Вычисление этого сходства будет служить также в качестве геометрической проверки петли.

Используется метод Horn основанный на $\text{sim}(3)$ группе алгебры ли.

Шаг 3: LOOP CLOSING

3) Loop Fusion

Объединили повторяющиеся точки и вставили новые рёбра в covisibility графе. Поправка преобразования распространяется на все узлы в петле. Для всех задействованных узлов запускается Track Local Map. Обновляются все задействованные рёбра в covisibility графе.

4) Essential Graph Optimization

Essential graph оптимизируется для коррекции дрейфа масштаба.



Используемые функции оптимизации

Решил не писать сюда функции оптимизации, но ежели вы хотите
То можем и их посмотреть



Пример работы ORB SLAM

Видосик не вставился. Свернуть и показать. Файл orb.ogv



LSD-SLAM (Large-scale direct monocular SLAM) – произошёл от метода SVO (Semi-Direct Monocular Visual Odometry). В качестве ключевого кадра предлагает изображение совмещенное с картой глубины полученной за счет выявления всех активных граней на карте интенсивности на обоих изображениях. Данный тип ключевых кадров позволяет данному методу строить карту с высокой плотностью точек на ней. Карта представлена в виде набора позиционных графов ключевых кадров. Основная проблема метода – фотометрическая ошибка.



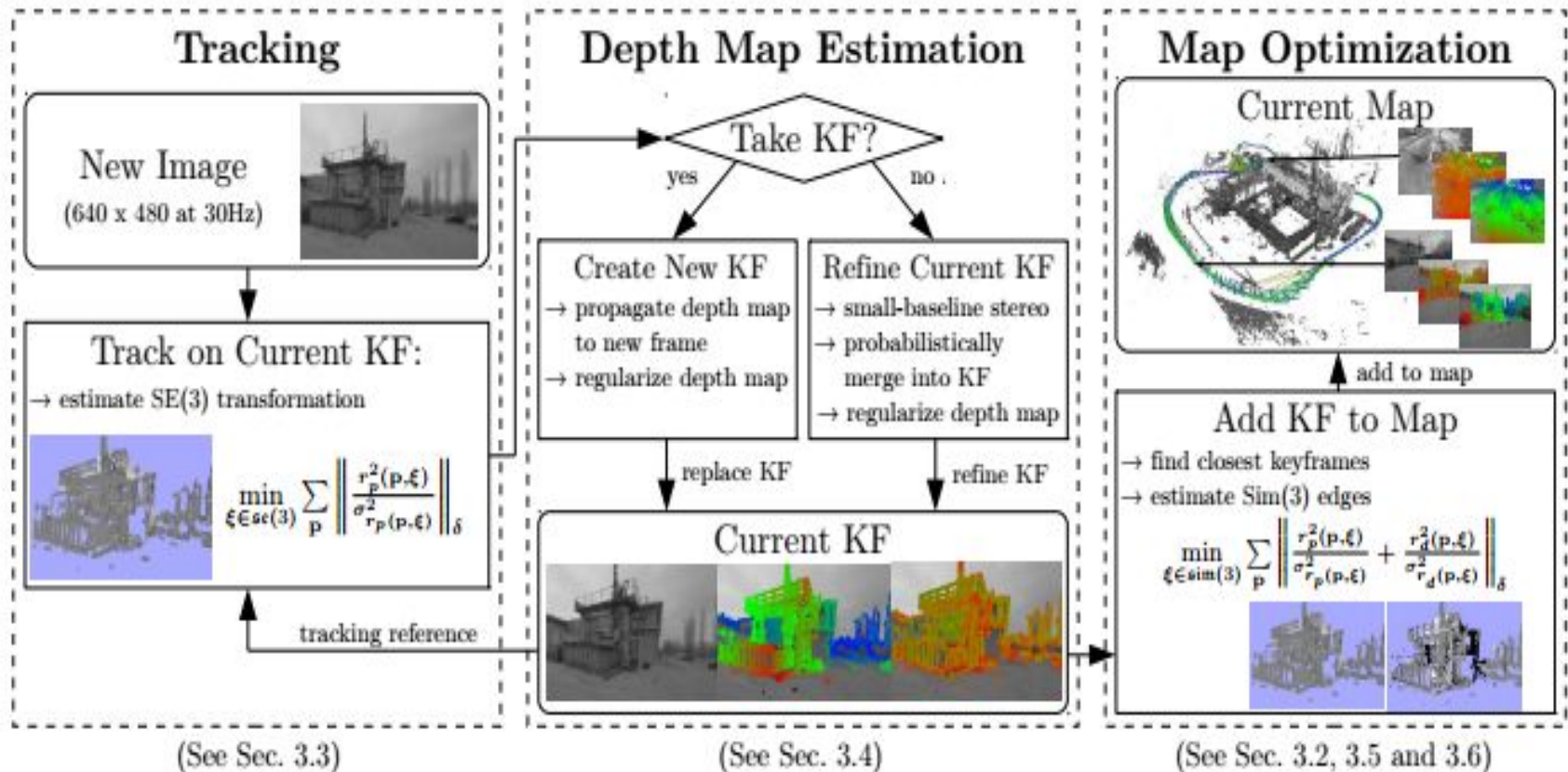


Fig. 3: Overview over the complete LSD-SLAM algorithm.

Используется алгебра Ли

Group	Description	Dim.	Matrix Representation
SO(3)	3D Rotations	3	3D rotation matrix
SE(3)	3D Rigid transformations	6	Linear transformation on homogeneous 4-vectors
SO(2)	2D Rotations	1	2D rotation matrix
SE(2)	2D Rigid transformations	3	Linear transformation on homogeneous 3-vectors
Sim(3)	3D Similarity transformations (rigid motion + scale)	7	Linear transformation on homogeneous 4-vectors

1) 3D трансформация твёрдого тела $G \in SE(3)$ ()

$$G = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad \text{with } \mathbf{R} \in SO(3) \text{ and } \mathbf{t} \in \mathbb{R}^3.$$

Позиции камеры соответствует элемент $\xi \in se(3)$. Элементы отображаются в $SE(3)$ с помощью экспоненциального отображения $G = \exp_{se(3)}(\xi)$, его обратный быть обозначенном $\xi = \log_{SE(3)}(G)$.

ξ_{ji} – преобразование точки от фрейма i к фрейму j

Операция контенация двух объектов: $se(3) \times se(3) \rightarrow se(3)$

$$\xi_{ki} := \xi_{kj} \circ \xi_{ji} := \log_{SE(3)} \left(\exp_{se(3)}(\xi_{kj}) \cdot \exp_{se(3)}(\xi_{ji}) \right).$$

Функция ω – проецирует точку p и её обратную глубину d в ξ фрейм камеры

$$\omega(p, d, \xi) := \begin{pmatrix} x'/z' \\ y'/z' \\ 1/z' \end{pmatrix} \quad \text{with} \quad \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} := \exp_{se(3)}(\xi) \begin{pmatrix} p_x/d \\ p_y/d \\ 1/d \\ 1 \end{pmatrix}$$

3D преобразование подобия $S \in Sim(3)$ – вращение, перенос, изменение масштаба.

$$S = \begin{pmatrix} sR & t \\ 0 & 1 \end{pmatrix} \quad \text{with} \quad R \in SO(3), \quad t \in \mathbb{R}^3 \quad \text{and} \quad s \in \mathbb{R}^+.$$

Два изображения считаются выравненными Гауссо-Ньютоновской оптимизацией по фотометрической ошибке при минимизации:

$$E(\xi) = \sum_i \underbrace{(I_{\text{ref}}(\mathbf{p}_i) - I(\omega(\mathbf{p}_i, D_{\text{ref}}(\mathbf{p}_i), \xi)))^2}_{=: r_i^2(\xi)},$$

Решаем используя левую композиционную формулировку:
Начинаем с оценки $\xi(0)$, в каждой итерации лево-умноженное увеличение на $\delta\xi(n)$ вычисляется путем решения для минимума Гаусса-Ньютона второго порядка аппроксимации E :

Не бойтесь, это лишь оптимизация гауса-ньютона в чистом виде для матриц!

$$\delta \xi^{(n)} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{r}(\xi^{(n)}) \quad \text{with} \quad \mathbf{J} = \left. \frac{\partial \mathbf{r}(\epsilon \circ \xi^{(n)})}{\partial \epsilon} \right|_{\epsilon=0},$$

где $\mathbf{J}$ является производной от сложенного остаточного вектора $\mathbf{r} = (r_1, \dots, r_n)^T$ по отношению к приращению лево-умноженного $\mathbf{J}^T \mathbf{J}$ является гауссо-ньютоновым приближением матрицы Гессе $\mathbf{E}$. Новая оценка получается увеличением вычисленного обновления

$$\xi^{(n+1)} = \delta \xi^{(n)} \circ \xi^{(n)}.$$

Чтобы избавиться от выбросов возникающих, например, при преграждение видимости камеры или отражения используется взвешённая схема. В каждой итерации, весовая матрица $W=W(\xi^{(n)})$, вычисляет, какие понижающего весов крупные остатки. Итеративно решена функция ошибки:

$$E(\xi) = \sum_i w_i(\xi) r_i^2(\xi),$$

И обновление вычисляется как:

$$\delta \xi^{(n)} = -(\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{W} \mathbf{r}(\xi^{(n)}).$$

Если предположить, что остатки независимые, то обратный Гессинан от $(\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1}$ является ковариационной Σ_ξ левого умножения ошибки на конечный результат, то есть:

$$\xi^{(n)} = \epsilon \circ \xi_{\text{true}} \text{ with } \epsilon \sim \mathcal{N}(0, \Sigma_\xi).$$

Распространение неопределенности представляет собой статистический инструмент для получения неопределенности выходе функции $f(X)$, из-за неопределенности на ее входе X . Предполагая, что X будет гауссово распределение с ковариационной Σ_X , ковариации $f(X)$ можно приблизить (с использованием якобиана J_f для f) по

$$\Sigma_f \approx J_f \Sigma_X J_f^T.$$

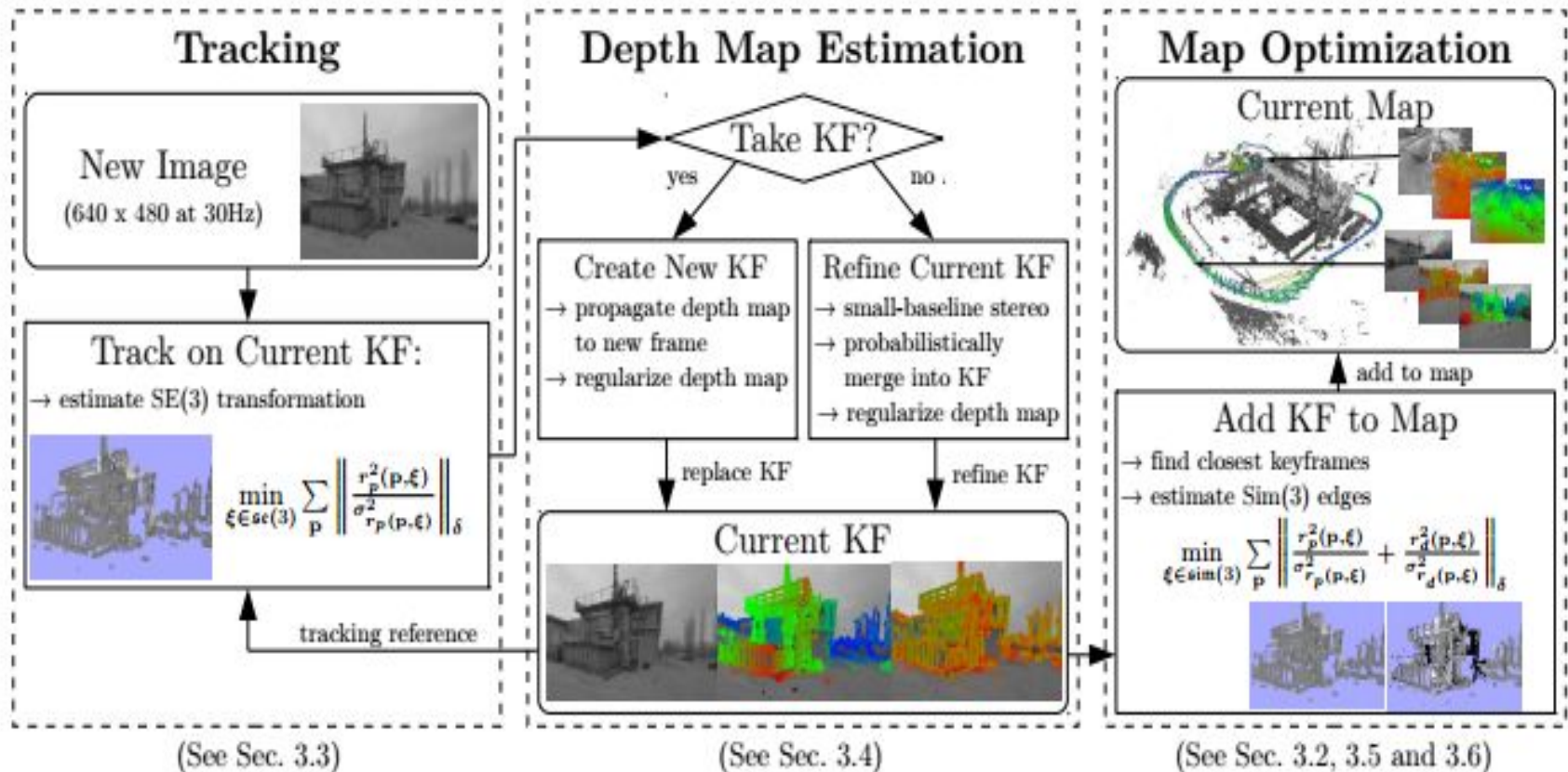


Fig. 3: Overview over the complete LSD-SLAM algorithm.

Представление карты

Карта представлена в виде графа позиций ключевых кадров.

Каждый ключевой кадр K_i состоит из:

- 1) Изображения камеры $I_i : \Omega_i \rightarrow R$
- 2) Обратную карту глубины $D_i : \Omega_{Di} \rightarrow R+$
- 3) Дисперсия обратной глубины $V_i : \Omega_{Di} \rightarrow R+$

Карта глубины и дисперсия определяются только для подмножества точек $\Omega_{Di} \subset \Omega_i$, содержащая все области изображения вблизи достаточно большого градиента интенсивности, следовательно, полу-густой.

Гранями E_{ji} между ключевыми кадрами содержат их относительное выравнивание, как подобия преобразования $\xi_{ji} \in SIM(3)$, а также соответствующая ковариационная матрица Σ_{ji} .



Шаг 0: Инициализация

Достаточно инициализировать случайную карту глубины с большой дисперсией. Несколько первых ключевых кадров сведут карту глубины к верному значению.

Шаг 1 :Tracking new Frames: Direct se(3) Image Alignment

Начиная с предыдущего ключевого кадра $K_i = (I_i, D_i, V_i)$ относительная 3D позиция $\xi_{ji} \in \text{se}(3)$ нового изображения I_j вычисляется путем минимизации дисперсии нормированной фотометрические ошибки:

$$E_p(\xi_{ji}) = \sum_{p \in \Omega_{D_i}} \left\| \frac{r_p(\mathbf{p}, \xi_{ji})}{\sigma_{r_p(\mathbf{p}, \xi_{ji})}^2} \right\|_{\delta}$$

with $r_p(\mathbf{p}, \xi_{ji}) := I_i(\mathbf{p}) - I_j(\omega(\mathbf{p}, D_i(\mathbf{p}), \xi_{ji}))$

$$\sigma_{r_p(\mathbf{p}, \xi_{ji})}^2 := 2\sigma_I^2 + \left(\frac{\partial r_p(\mathbf{p}, \xi_{ji})}{\partial D_i(\mathbf{p})} \right)^2 V_i(\mathbf{p})$$

where $\|\cdot\|_{\delta}$ is the Huber norm

$$\|r\|_{\delta} := \begin{cases} \frac{r^2}{2\delta} & \text{if } |r| \leq \delta \\ |r| - \frac{\delta}{2} & \text{otherwise.} \end{cases}$$

Дисперсия невязки σ_I^2 вычисляется, как в Propagation of Uncertainty.

Используется гаусовское отображение интенсивности шума $\sigma_{r_p(\mathbf{p}, \xi_{ji})}^2$

Минимизация осуществляется с использованием итеративного повторно взвешенной оптимизации Гаусса-Ньютона

Выбор ключевого кадра. Если камера перемещается слишком далеко от существующей карты, то новый ключевой кадр создаётся из последнего отслеживаемого изображения. Пороговая оценка возведённого сочетания относительно расстояния и угла относительно текущего ключевого кадра текущего.

$$\text{dist}(\xi_{ji}) := \xi_{ji}^T W \xi_{ji}$$

Создание карты глубины. После того, как новый ключевой кадр выбран чтобы стать новым ключевым кадром, его карта глубины инициализируется путём проецирования точек из предыдущего ключевого кадра на новый с помощью SVO, а затем за одну итерацию пространственной регуляризации и удаления аномальных значений. Затем карта глубины масштабируется, чтобы иметь среднюю обратную глубину равную единицы – этот коэффициент масштабирования непосредственно включается в $\text{sim}(3)$ позицию камеры. Далее происходит смена ключевых кадров.

Детализация карты глубины. Отслеживаемые кадры, которые не являются ключевыми используются для уточнения текущего ключевого кадра: Большое количество очень эффективных smallbaseline стерео сравнений выполняется для областей изображения, где ожидаемая точность стерео достаточно велика, как это описано в SVO. В результате включается в существующую карту глубины, таким образом, ее переработки и потенциально добавляя новые пиксели - это делается с помощью фильтрации, описанной в SVO.

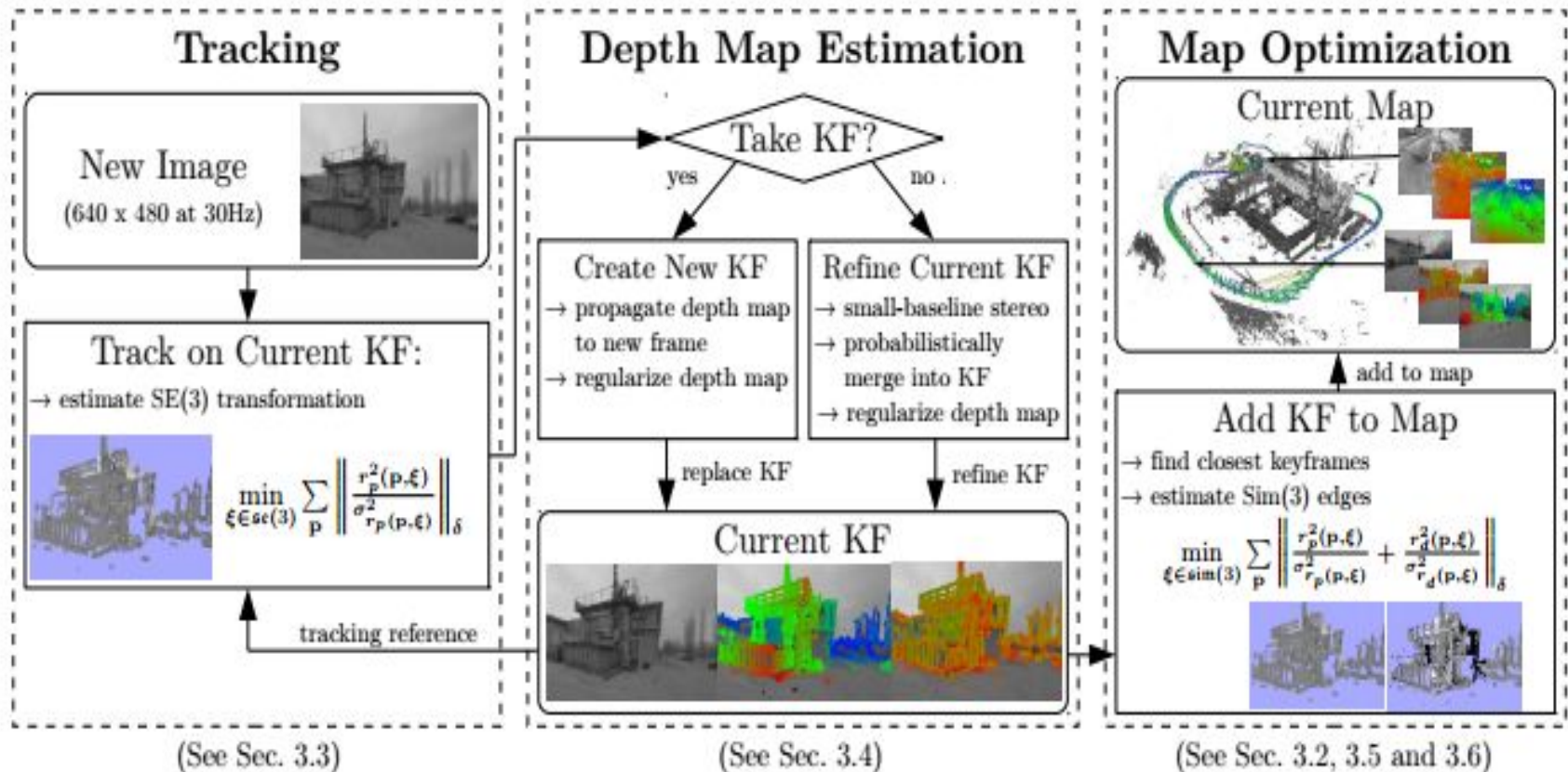


Fig. 3: Overview over the complete LSD-SLAM algorithm.

Direct Image Alignment on $\text{sim}(3)$.

Для решения проблемы дрейфа масштаба, используя присущую корреляцию между глубиной сцены и точности отслеживания: карта глубины каждого созданного ключевого кадра масштабируется таким образом, что средняя обратная глубина равна единице. В свою очередь, ребра между ключевыми кадрами оцениваются как элементы $\text{sim}(3)$, элегантно включающий разницу масштабирования между ключевыми кадрами, и, в частности, для больших петель-замыканий, что позволяет явное обнаружение накопленного масштабного сноса.

Шаг 3: Constraint Acquisition: Direct sim(3) Image Alignment

Они предложили новый метод выравнивания двух разномасштабных ключевых кадров на SIM(3). В дополнение к фотометрической ошибке r_p они включают остаточную глубину r_d , которая отражает отклонение в обратной глубине между ключевыми кадрами, что позволяет непосредственно оценить масштабированное преобразование между ними. Общая функция ошибки, которая сведена к минимуму становится:

$$E(\xi_{ji}) := \sum_{\mathbf{p} \in \Omega_{D_i}} \left\| \frac{r_p^2(\mathbf{p}, \xi_{ji})}{\sigma_{r_p}^2(\mathbf{p}, \xi_{ji})} + \frac{r_d^2(\mathbf{p}, \xi_{ji})}{\sigma_{r_d}^2(\mathbf{p}, \xi_{ji})} \right\|_{\delta},$$

Остаточная глубина и её дисперсия вычисляется как:

$$\begin{aligned} r_d(\mathbf{p}, \xi_{ji}) &:= [\mathbf{p}']_3 - D_j([\mathbf{p}']_{1,2}) \\ \sigma_{r_d(\mathbf{p}, \xi_{ji})}^2 &:= V_j([\mathbf{p}']_{1,2}) \left(\frac{\partial r_d(\mathbf{p}, \xi_{ji})}{\partial D_j([\mathbf{p}']_{1,2})} \right)^2 + V_i(\mathbf{p}) \left(\frac{\partial r_d(\mathbf{p}, \xi_{ji})}{\partial D_i(\mathbf{p})} \right)^2 \end{aligned}$$

$$\text{where } \mathbf{p}' := \omega_s(\mathbf{p}, D_i(\mathbf{p}), \xi_{ji})$$

Constraint Search.

После того как новый ключевой кадр добавляется в ряд $K_{j1}, \dots, K_{jn}$ проверяются возможные замыкания петли. Используют 10 самых близких ключевых кадров, а также подходящего кандидата, предложенные появление на основе алгоритма отображения [11] для обнаружения замыкания крупномасштабных петель. Чтобы избежать ложных или ложно отслеживаемых замыканий цикла выполняется взаимная проверка отслеживания: Для каждого кандидата K_{jk} независимо отслеживается ξ_{jki} и ξ_{ijk} и они должны быть статически подобны, чтобы добавить новую связь на карту. Данная величина должна быть небольшой:

$$e(\xi_{jki}, \xi_{ijk}) := (\xi_{jki} \circ \xi_{ijk})^T \left(\Sigma_{jki} + \text{Adj}_{jki} \Sigma_{ijk} \text{Adj}_{jki}^T \right)^{-1} (\xi_{jki} \circ \xi_{ijk})$$

После нахождения замыкания цикла

Adj_{jki} is used to transform Σ_{ijk} into the correct tangent space.

Шаг 4: Map optimization

Карта, состоящая из набора ключевых кадров и отслеживаемых $\text{sim}(3)$ – ограничений, непрерывно оптимизируется в фоновом режиме, используя позу оптимизации графика [18]. Функция ошибки, которая сведена к минимуму это:

$$E(\xi_{W_1} \dots \xi_{W_n}) := \sum_{(\xi_{ji}, \Sigma_{ji}) \in \mathcal{E}} (\xi_{ji} \circ \xi_{W_i}^{-1} \circ \xi_{W_j})^T \Sigma_{ji}^{-1} (\xi_{ji} \circ \xi_{W_i}^{-1} \circ \xi_{W_j}).$$



Пример работы LSD

Опять не тот формат видеосика. Lsd.ogv

DSO (Direct Sparse Odometry) – алгоритм визуальной одометрии на основе монокамеры. Использует минимизацию фотометрической ошибки, но использует неплотную карту.

Не переживайте, тут я устал и решил кратко рассказать!

Не дано мне видео вставлять в презентацию. =(

RTAB-Map - <https://introlab.github.io/rtabmap/>

ORB - <https://arxiv.org/pdf/1502.00956.pdf>

LSD -

https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwii66yzsuXQAhWG2CwKHXLzDkoQFggjMAE&url=https%3A%2F%2Fvision.in.tum.de%2F_media%2Fspezial%2Fbib%2Fengel14eccv.pdf&usg=AFQjCNFXD8s5MGfd4IBcRFHaykkdw2TgXg&sig2=D7RQomXpGH6pY7f9o20ZrA&bvm=bv.141320020,d.bGg

DSO - <https://arxiv.org/pdf/1607.02565.pdf>

Спасибо за внимание!