Курс «Трёхмерное компьютерное зрение»

Тема №4

«Разреженная 3D реконструкция и SLAM»

Антон Конушин

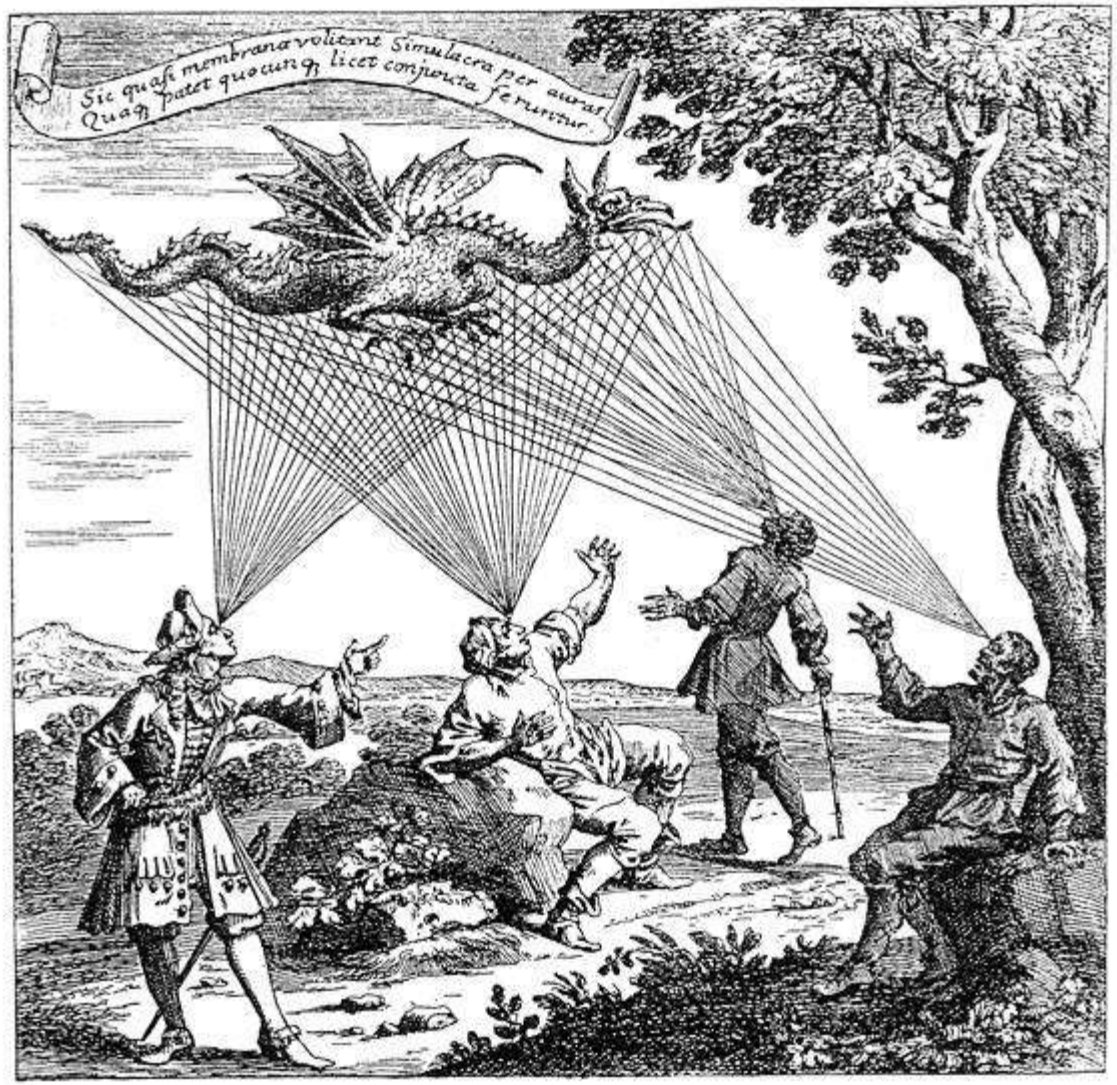
План

- Structure from Motion (SFM)
- BigSFM
- COLMAP
- SLAM
- Классический подход к SLAM
- Нейросетевой подход к SLAM

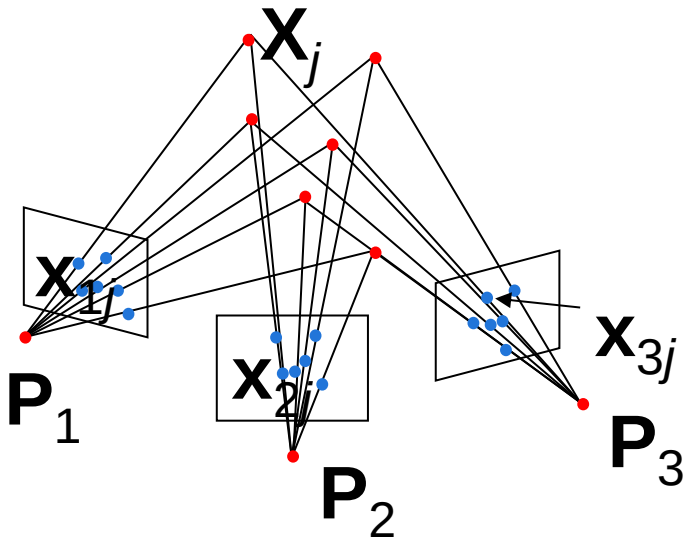


Смысл задачи
Structure from Motion

Структура из движения



Драконъ, видимый подъ различными углами зрѣнія
По гравюрѣ на мѣди изъ „Oculus artificialis teledioptricus" Цана. 1702 года.



Движение проекций 3D точек, вызванное движением камеры, позволяет определить координаты 3D точек (*структуру, structure*) и само движение камеры (*движение, motion*).

Это задача «структура из движения» (Structure from Motion, SfM)

Пример результата работы SfM

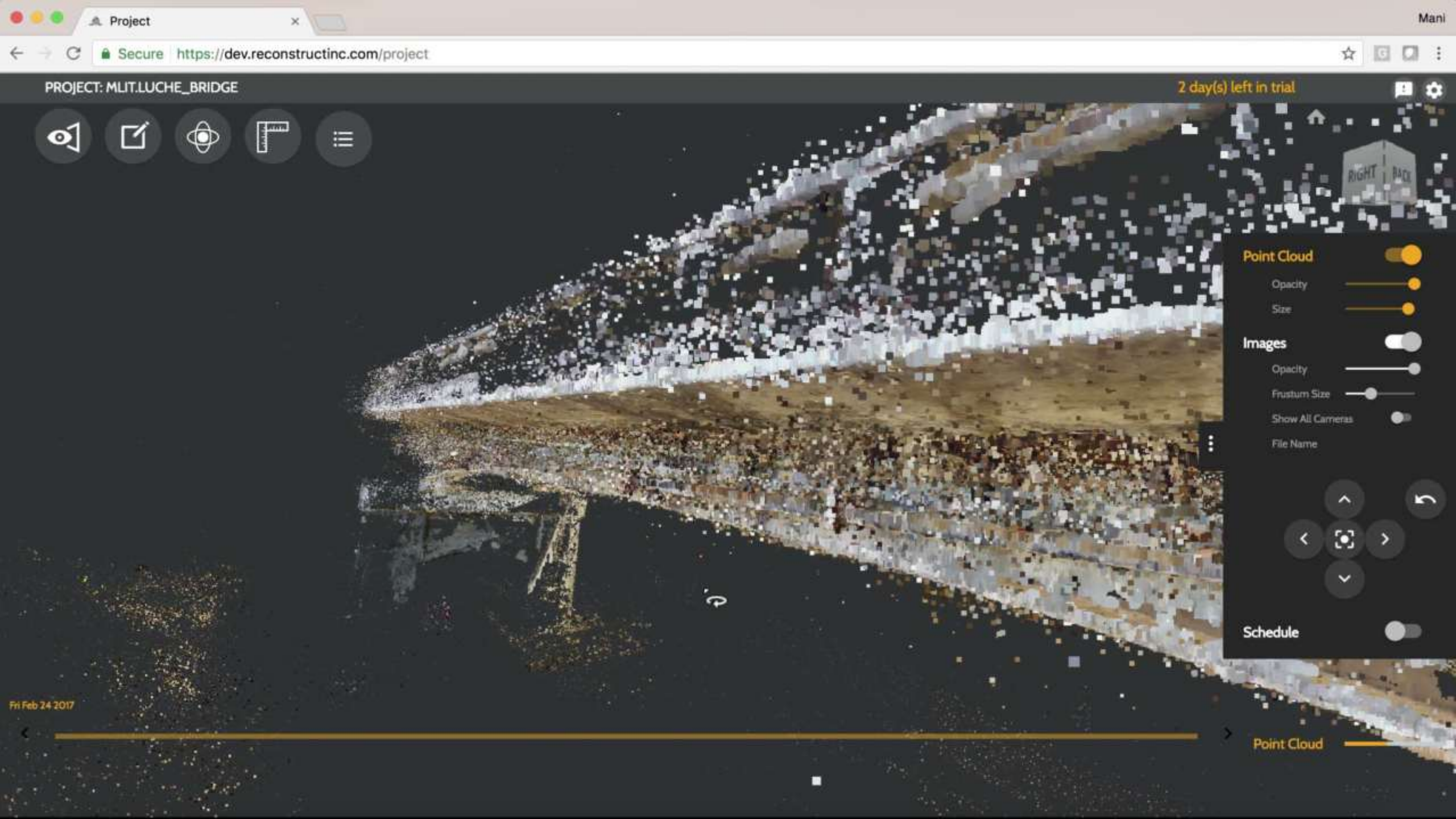


Пример приложения

Инспекция в труднодоступных местах по съёмке с дронов

- Строим 3D модель по изображениям
- Даём инструменты для визуального изучения изображения и отображения взаимодействия в 3D




Project
Mani
Secure https://dev.reconstructinc.com/project
PROJECT: MLIT.LUCHE_BRIDGE
2 day(s) left in trial
RIGHT
BACK
Point Cloud
Opacity
Size
Images
Opacity
Frustum Size
Show All Cameras
File Name
Schedule
Fri Feb 24 2017
Point Cloud

Зачем ещё нужна SfM?



- Первый шаг в решении других 3D задач
- Калибровка камер для плотной 3D реконструкции
- Калибровка камер для применения NeRF / 3DGS

Схема разреженной 3D реконструкции по изображениям

3D реконструкция по изображениям



Набор изображений

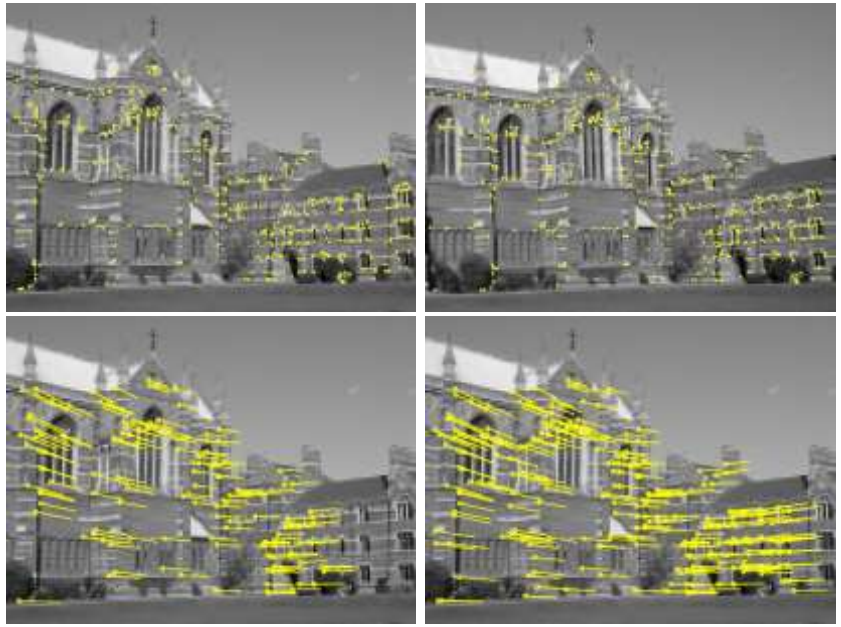


Трёхмерная (3D) модель

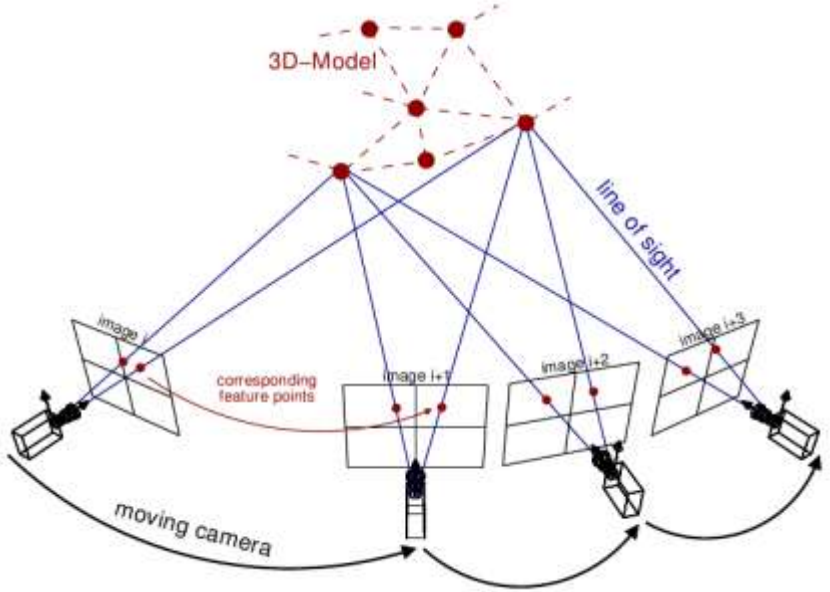
Пример задачи: построение 3D модели реального объекта
по набору фотографий из интернета

Source: Agarwal et. al. Building Rome in a Day. ICCV 2009

Разреженная реконструкция



Point features and matching

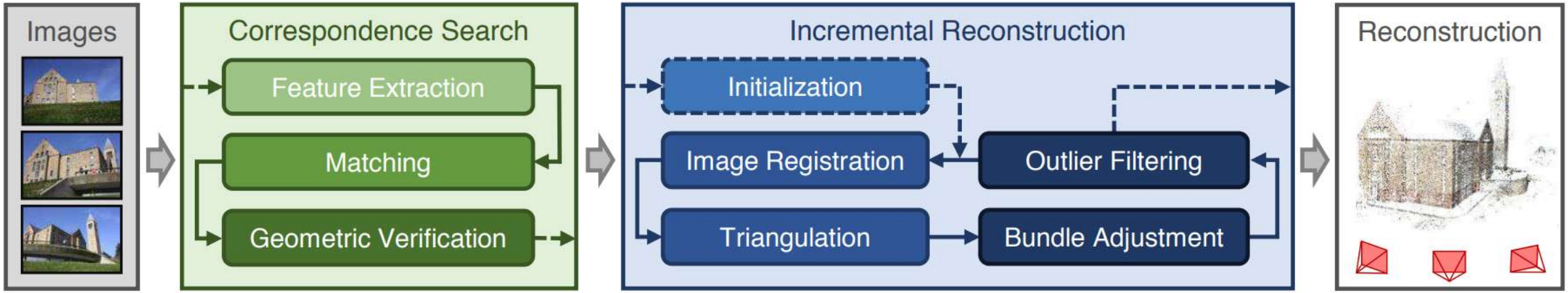


Camera localization and structure estimation

- Опираемся на сопоставление ключевых точек между изображениями
- 3D реконструкция в форме облака точек, соответствующих найденным ключевым точкам на изображениях
- Решаем совместно определение 3Д точек, соответствующих ключевым, и положение камер в пространстве относительно них
- 3D реконструкция – обратная задача к задаче компьютерной графики

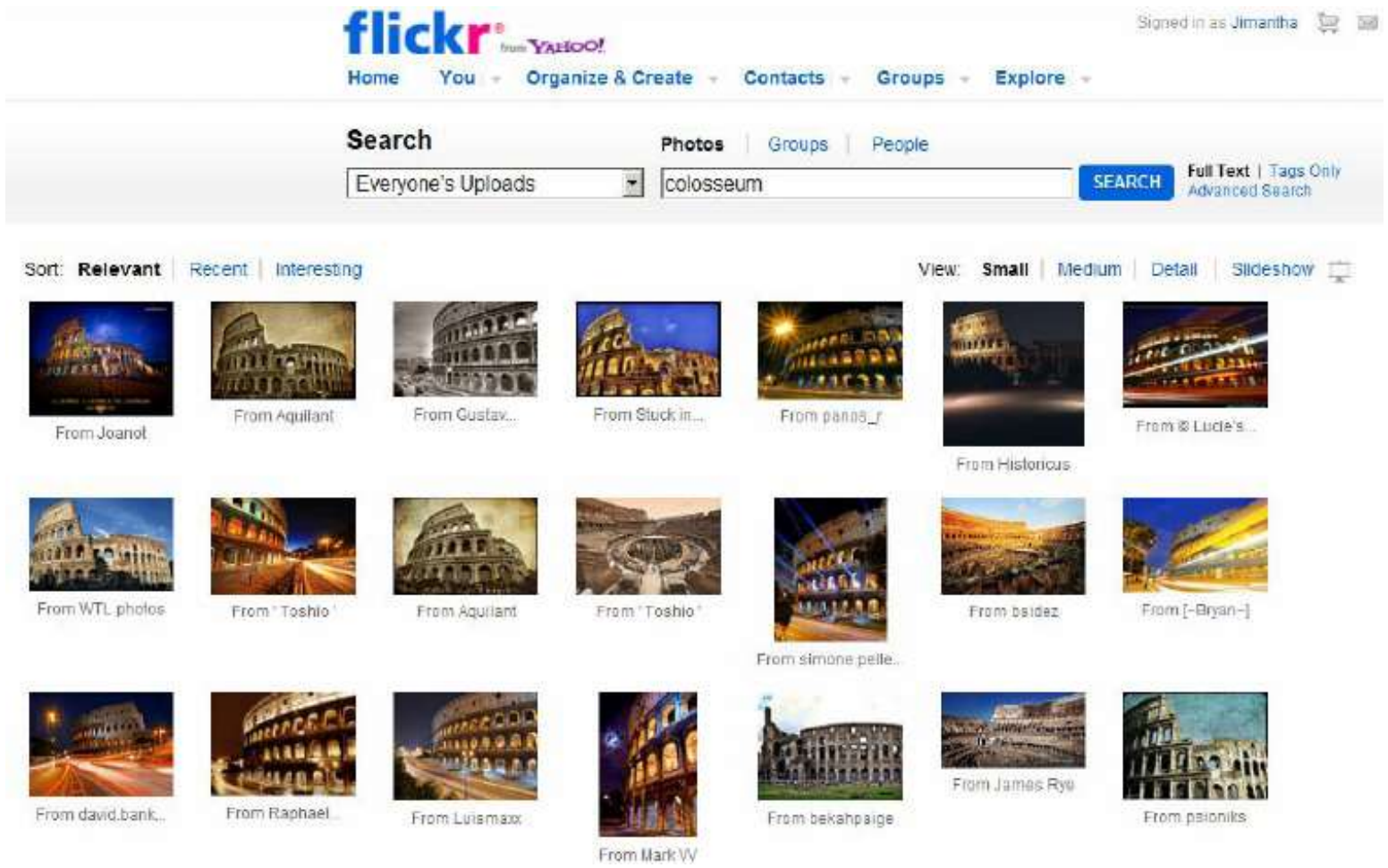
Инкрементальный подход

Классический инкрементальный подход, в котором изображения по одному добавляются в модель до полного перебора всех изображений



Source: Structure-From-Motion Revisited. CVPR2016

Коллекции изображений



Несколько миллионов изображений по тегу «Rome»

Обычные фотографии

- Получены разными пользователями с разных фотоаппаратов



iPhone 3G



Nikon D3

Инициализация внутренней калибровка



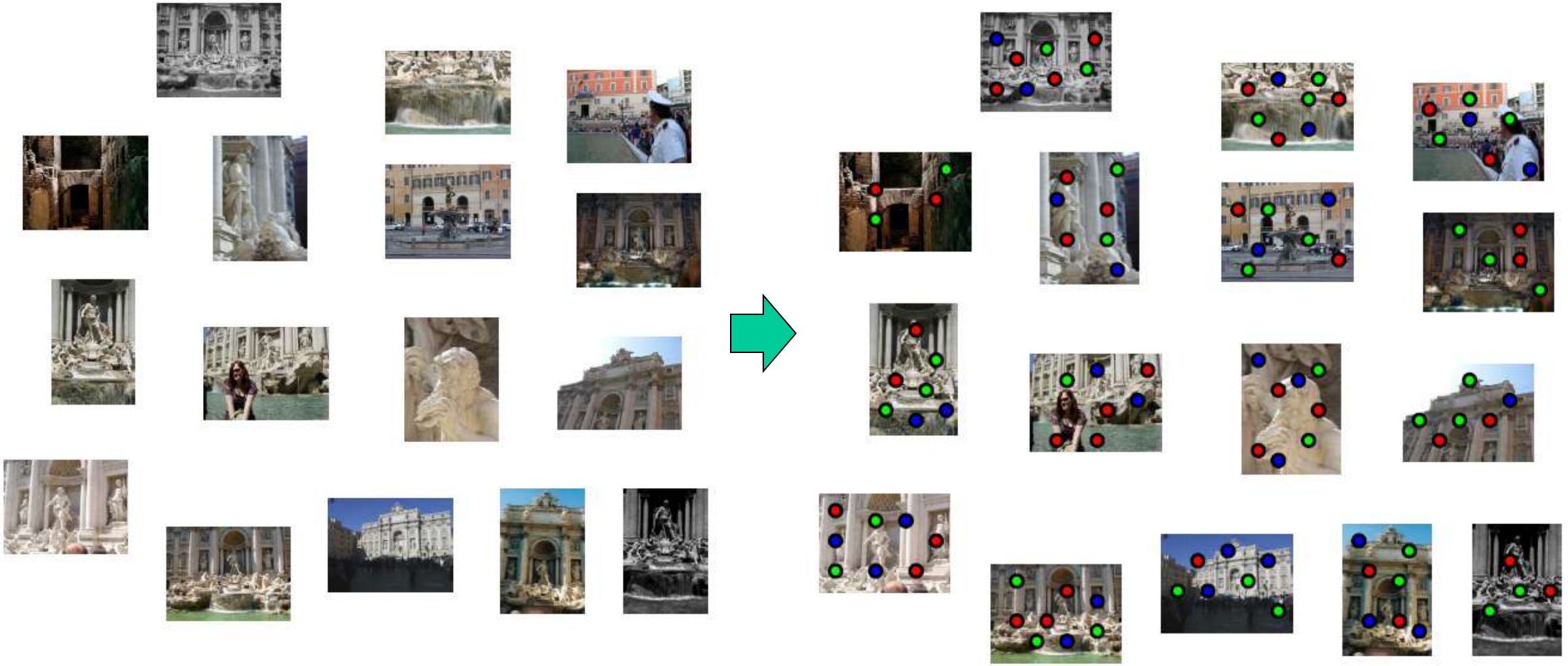
```
File size     : 85111 bytes
File date     : 2005:12:16 04:17:12
Camera make   : Panasonic
Camera model  : DMC-FZ20
Date/Time     : 2005:03:19 12:52:33
Resolution    : 450 x 600
Flash used    : No
Focal length  : 6.0mm
Exposure time: 0.0012 s  (1/800)
Aperture      : f/5.6
ISO equiv.    : 80
Whitebalance : Auto
Metering Mode: matrix
Exposure      : program (auto)
```

Структура EXIF-данных

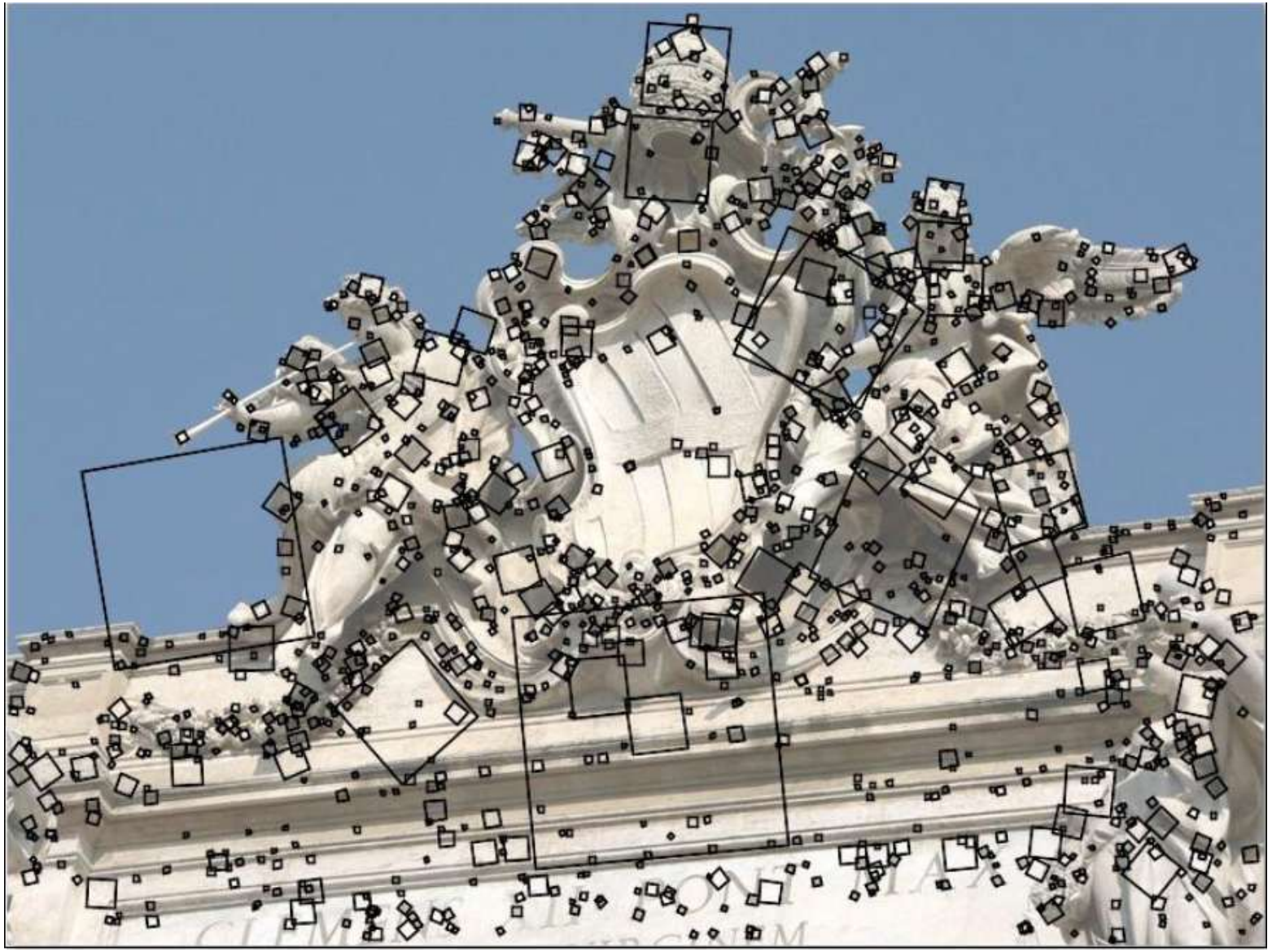


Матрица из спецификации камер

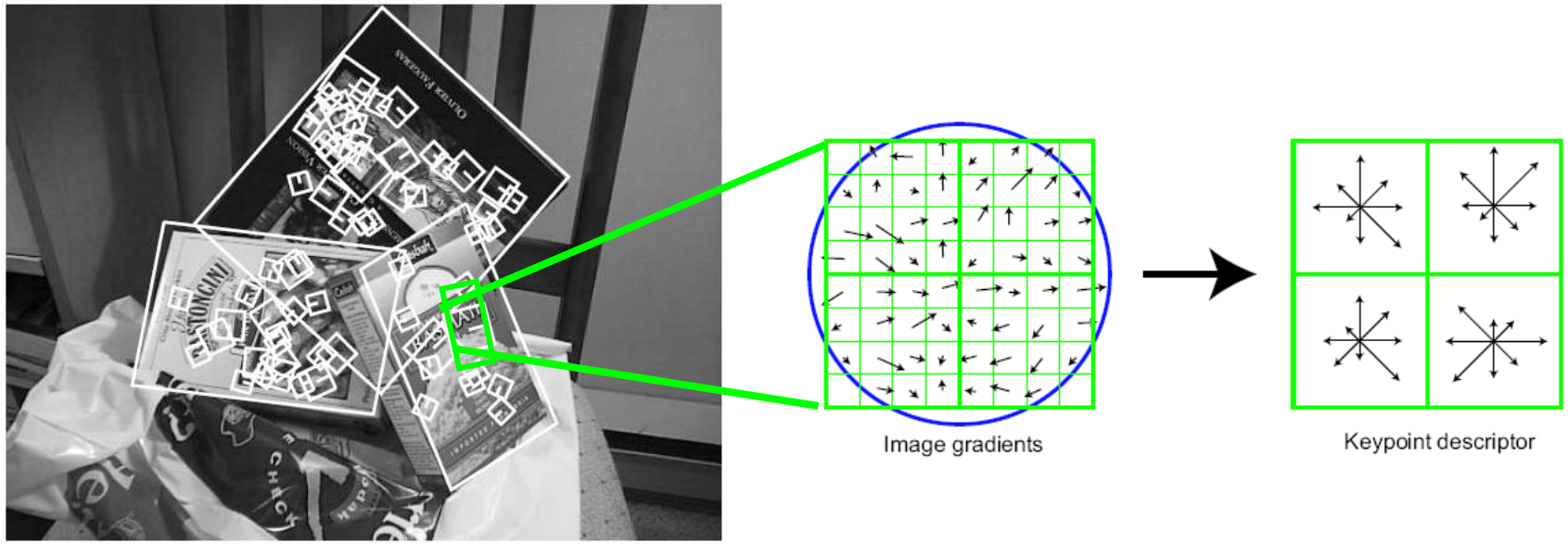
Найдем соответствующие точки



Найденные особые точки



Локальные дескрипторы: SIFT Descriptor



Histogram of oriented gradients (HoG)

- Описываем текстуру в окрестности
- Устойчивы к небольшим смещениям и аффинным преобразованиям

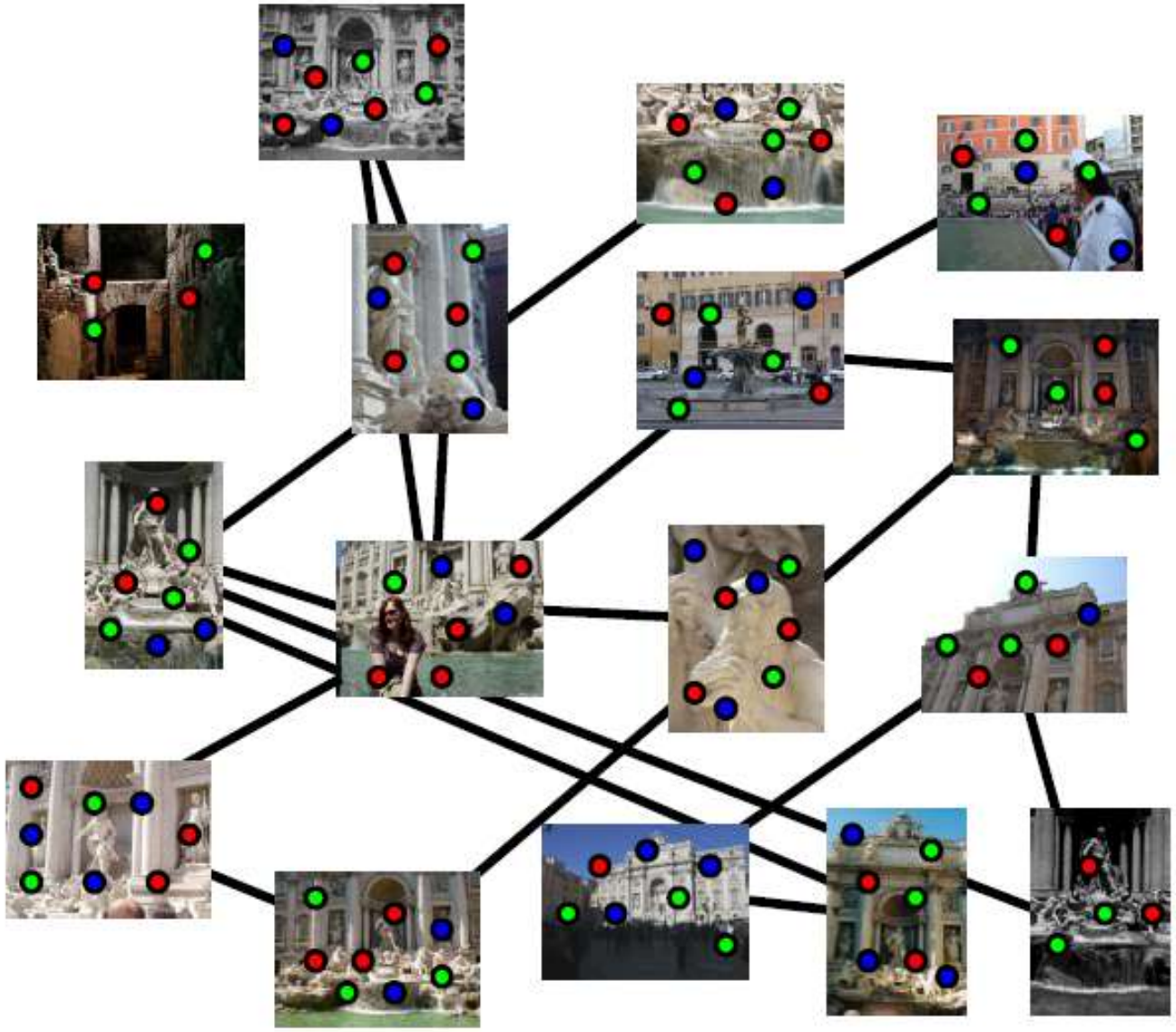
[Lowe, ICCV 1999]

Сопоставление изображений



- Сопоставляем точки по дескрипторам
- Фильтруем ложные соответствия по эпиполярной геометрии
- Для этого будем с помощью RANSAC вычислять фундаментальную матрицу
- Пары, между которыми много соответствий, будем считать «связанными»

Граф связанности изображений



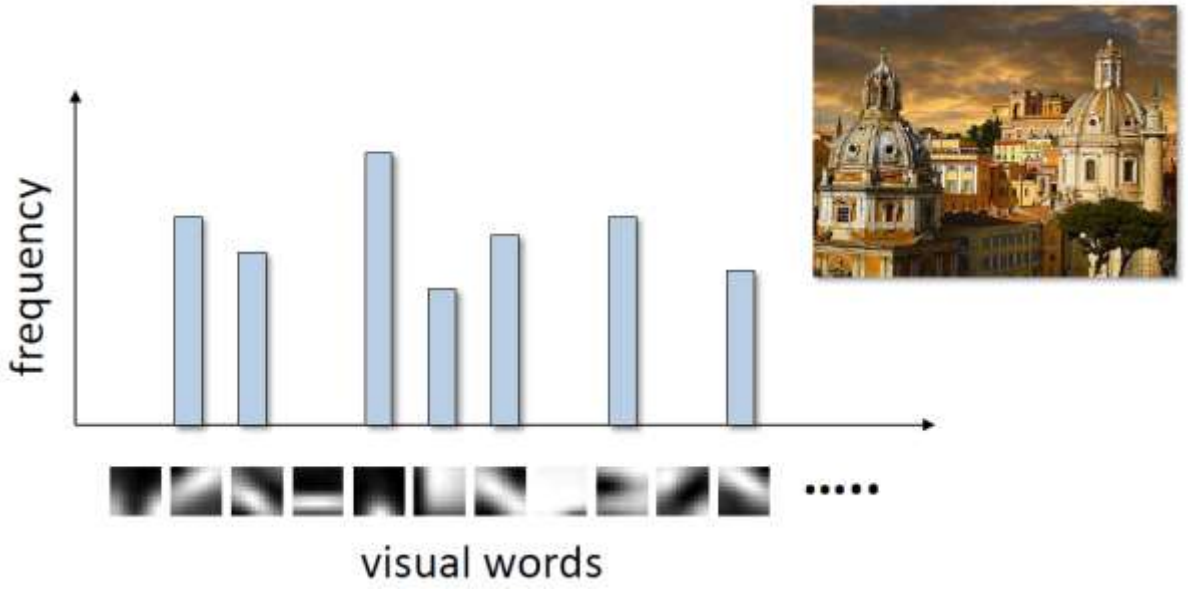
Объем задачи



- 250К изображений -> 31М пар изображений
- 2 пары изображений в секунду
  - 1 год на 500 машинах
- 1М изображений -> 500 000М пар изображений
  - 15 лет на 500 машинах

Поиск похожих изображений

- Метод «Мешок слов»
  - •Обучаем словарь визуальных слов размером 100К
  - •Строим гистограмму частот для каждого изображения:



  - • Для каждого изображения ищем 40 наиболее похожих
  - • Сопоставляем точки между этими парами изображений
  - • Выбираем хорошие пары

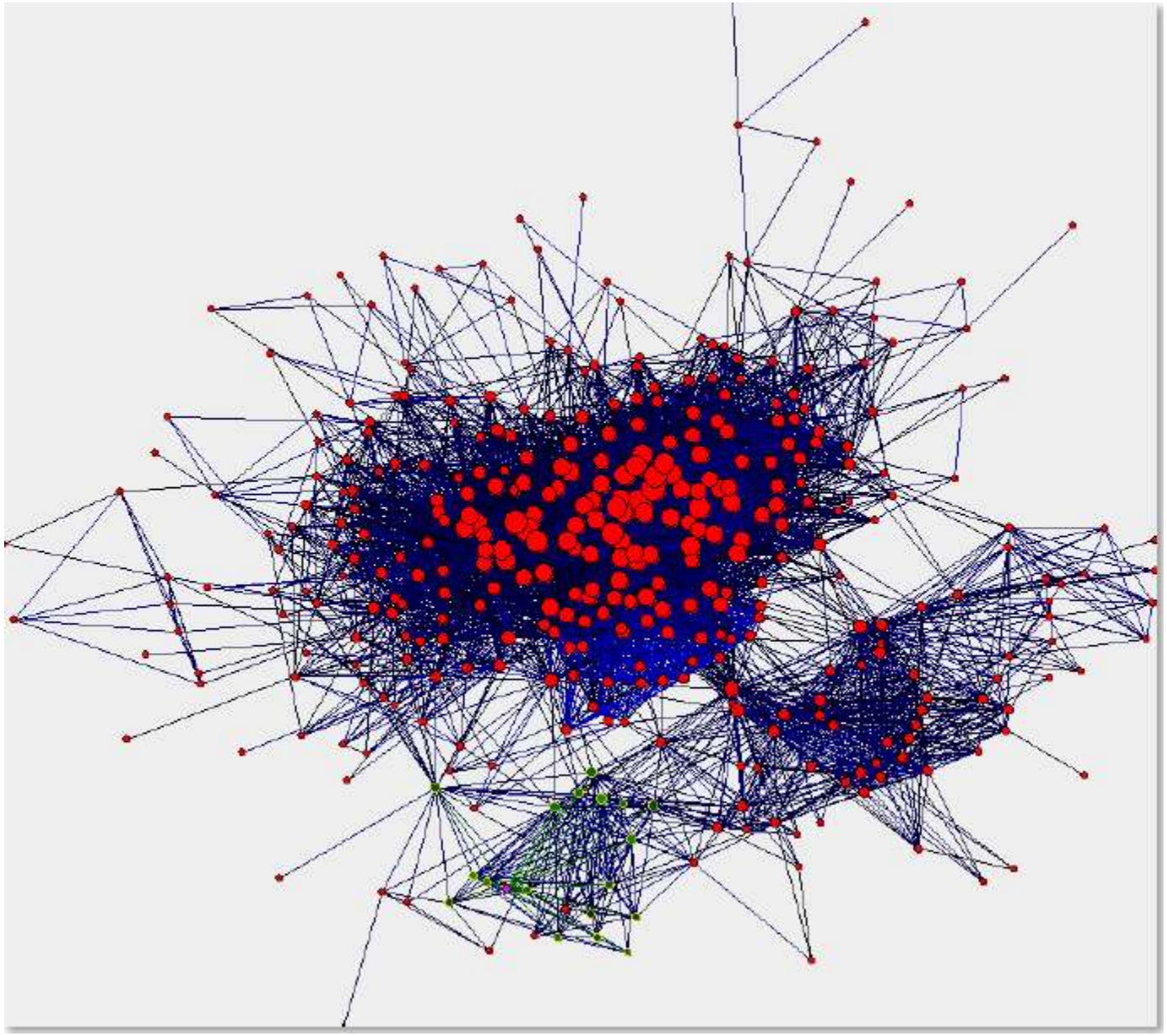
Результат сопоставлений



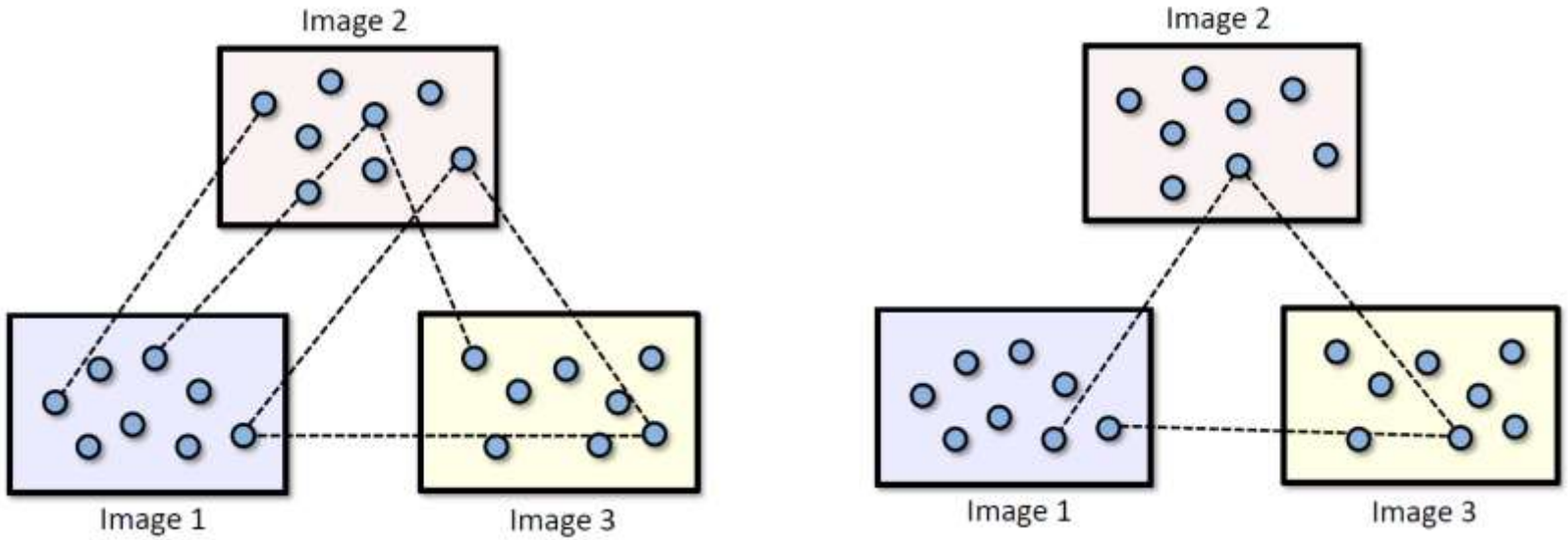


Мощность дескрипторов SIFT!

Граф связанности изображений

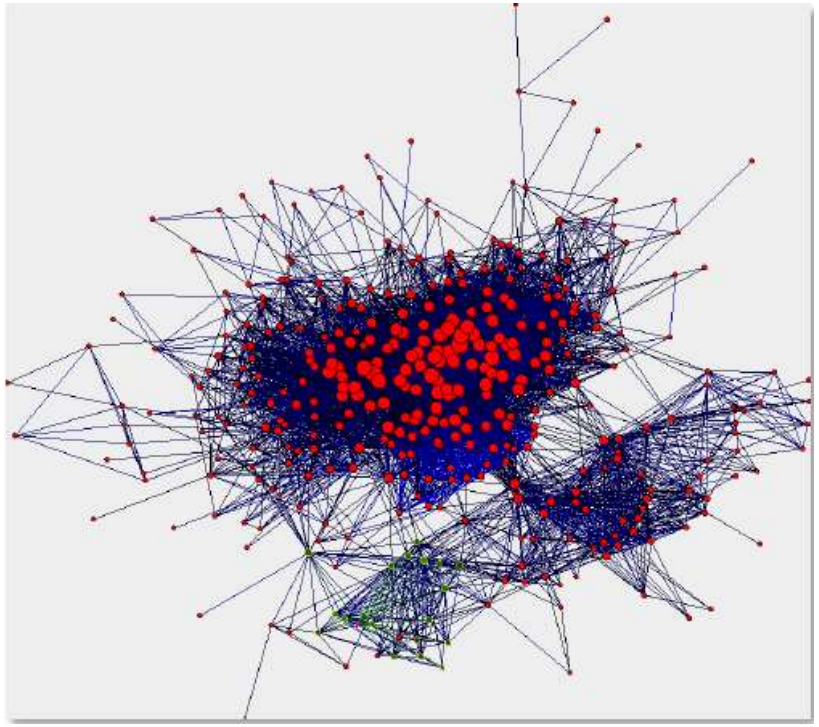


Построение «следов»

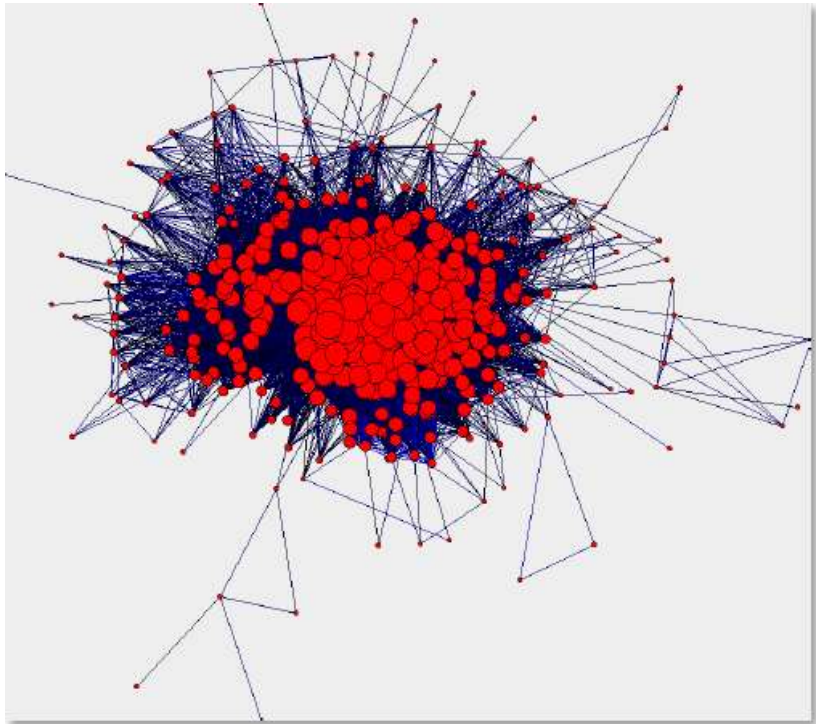


- Объединяем соответствия в «следы»
  - Отбрасываем незамкнутные следы, как потенциальные ошибки
- Пример (3000 изображений), всего 1.5M+ следов
  - 79% длины 2
  - 90% длины <=3
  - 98% длины <= 10
  - Самый длинный след 385 точек

Граф связанности изображений



Исходные ссоответствия

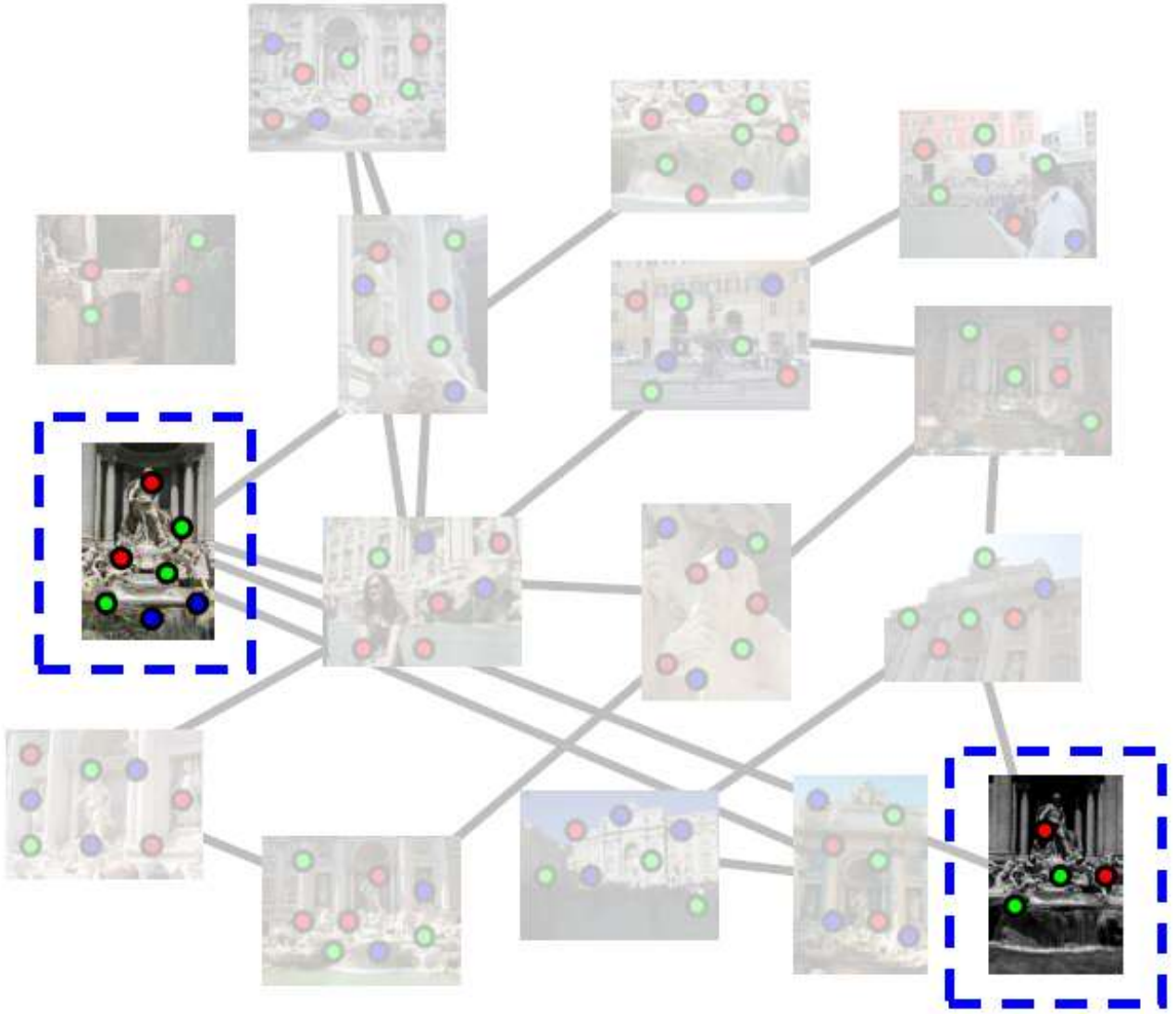


После построения следов

Стандартная схема

- Выбор опорных пар изображений
- Вычисление структуры по паре изображений
  - Построение матрицы внутренней калибровки
  - Вычисление фундаментальной и существенной матрицы
  - Триангуляция точек
  - Уточнение методом связок
- Добавление новых изображений
  - Калибровка камеры по известным 2D/3D соответствиям
  - Триангуляция новых точек
  - Уточнение методом связок

Выбор опорной пары



Выбор опорной пары




(+) много точек

(-) малая база




(+) большая база

(-) мало точек




(+) много точек

(+) большая база

Выбор опорной пары

- Возможны разные эвристики
- Гомография плохо описывает сцену, когда сцена не плоская, и существенный параллакс (большая база)
- Пример:
  - Возьмем пары, в которых > 100 соответствий
  - Вычислим гомографию и фундаментальную матрицу
  - Выберем пару изображений, минимизирующее отношение:

$$\frac{\mathrm{numInliers}_{\mathrm{homography}}}{\mathrm{numInliers}_{\mathrm{Fmatrix}}}$$

Примеры реконструкций



Дубровник



Примеры реконструкций





BigSFM

BigSFM

BigSFM: Reconstructing the World from Internet Photos

Our group is working on building a 3D model of the world from online photo collections, and our research spans several areas, from image features, to large-scale image matching, to structure-from-motion optimization, to applications such as location recognition. This page summarizes our work, has links to **code** and **datasets** we have made available, and has a description of each **project**.

Cornell Participants (and alumni): Noah Snavely (Cornell), David Crandall (Indiana University), Daniel Hauagge (Cornell), Kyle Wilson (Cornell), Song Cao (Cornell), Yin Lou (Cornell), Yunpeng Li (EPFL), Andrew Owens (MIT), Johannes Gehrke (Cornell) Dan Huttenlocher (Cornell)

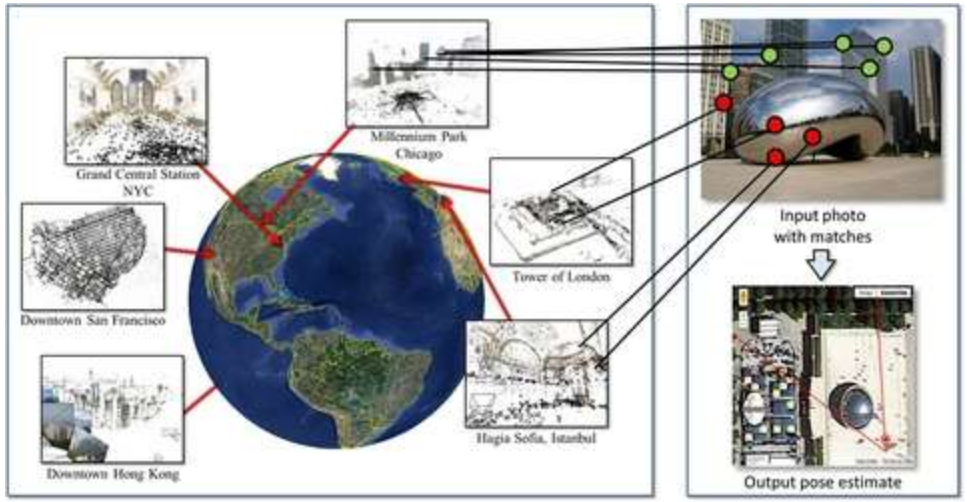
Other Collaborators: Sameer Agarwal (University of Washington and Google), Brian Curless (UW), Yasutaka Furukawa (Google), Steve Seitz (UW), Ian Simon (UW), Rick Szeliski (Microsoft Research)

We gratefully acknowledge the support of the following sponsors:

National Science Foundation | Amazon Web Services in Education | MIT Lincoln Labs | Microsoft | IU Data to Insight Center

http://www.cs.cornell.edu/projects/bigsfm/

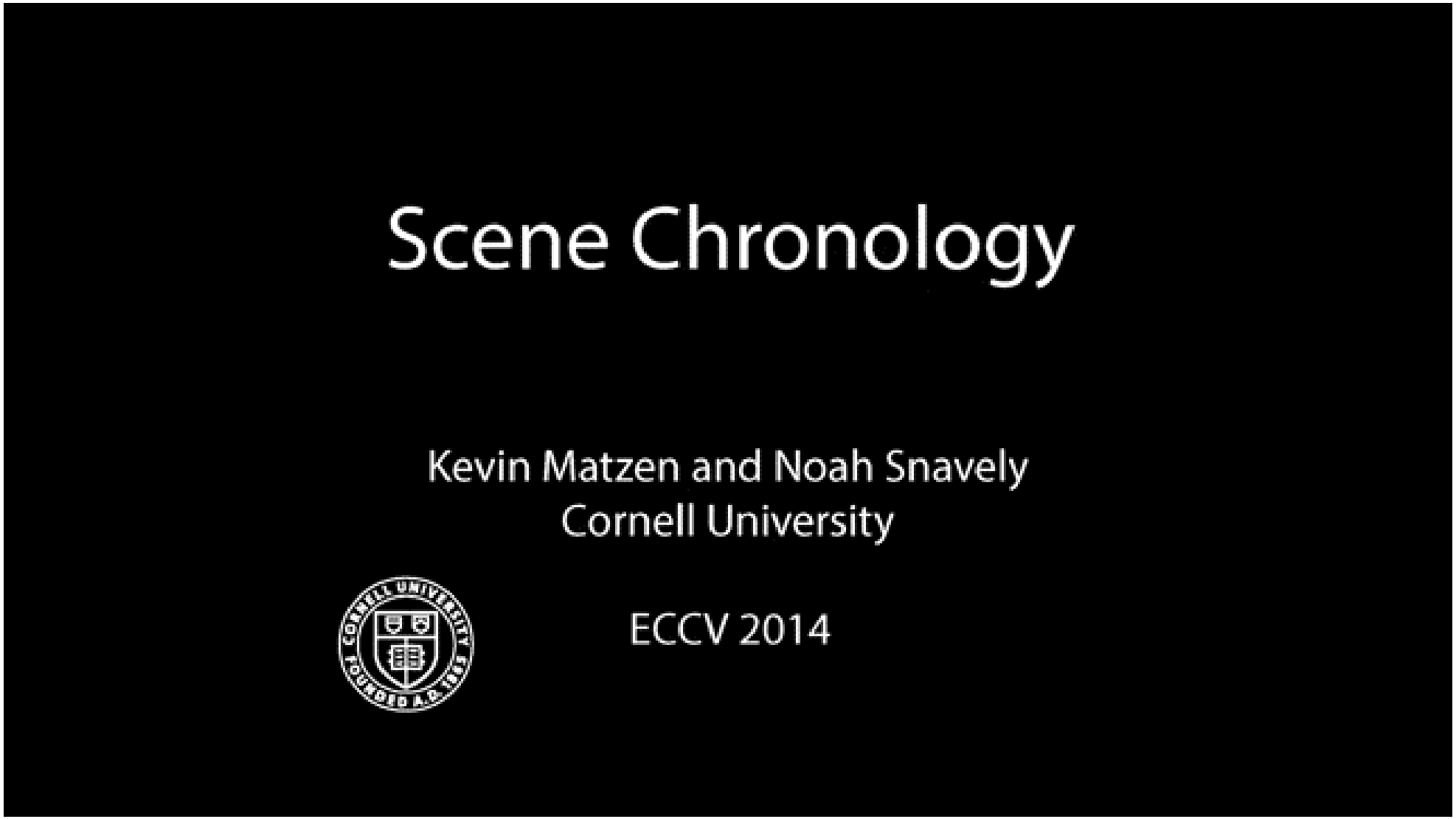
Глобальное позиционирование



Ищем похожие фотографии + сопоставляет по ключевым особенностям с 3д точками



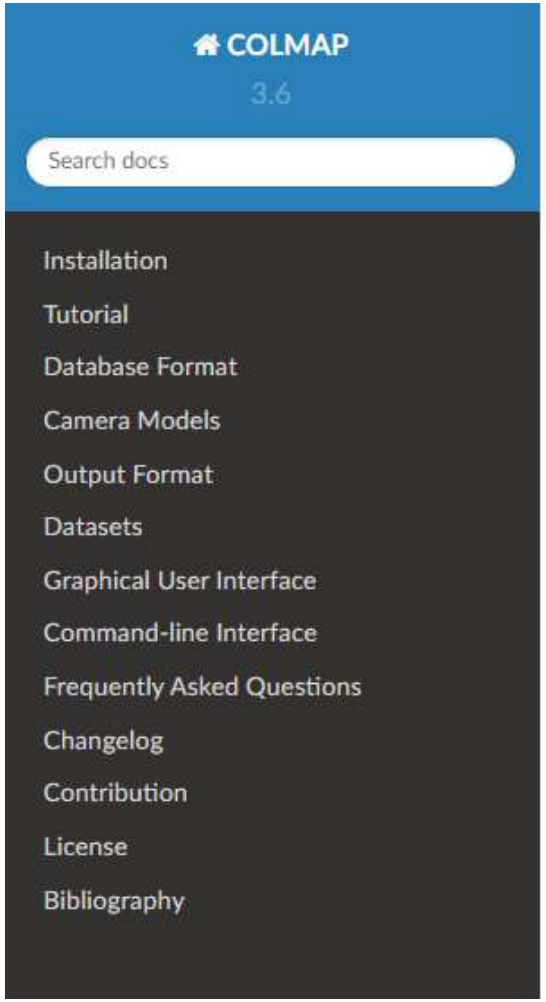
Визуальная хронология





COLMAP

COLMAP



Docs » COLMAP View page source

COLMAP



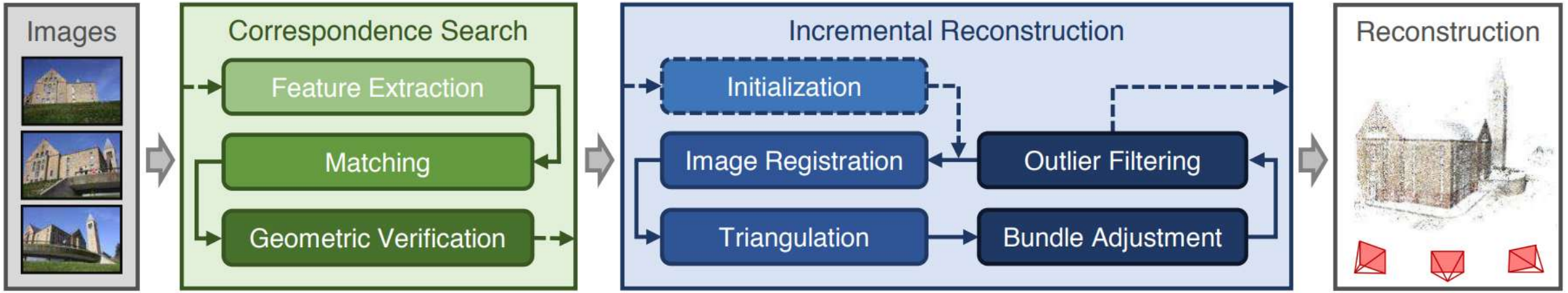
Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.



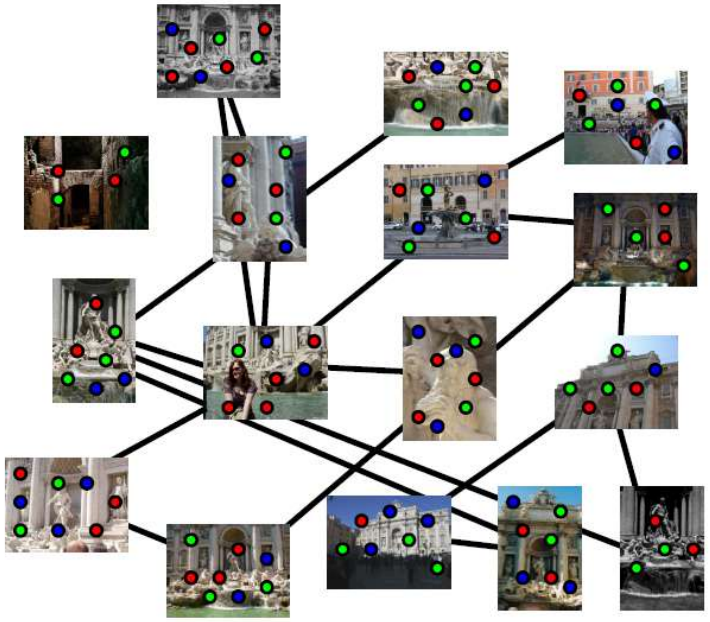
Dense models of several landmarks produced by COLMAP's MVS pipeline.

https://colmap.github.io/

Улучшение графа сцены

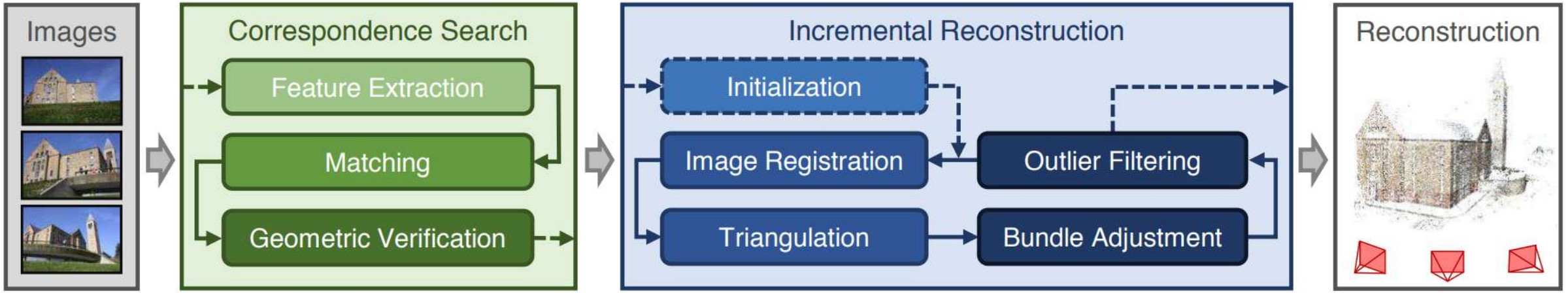


Source: Structure-From-Motion Revisited. CVPR2016



- Для калиброванных пар изображений считаем также существенную матрицу, движение камеры, триангулируем точки и a_m - медианный угол
- Если пара кадров хорошо описывается гомографией, то это позволит нам различать чистый поворот (панораму) и плоскую сцену
- Для точек на границе считаем преобразование подобия, и если доля таких точек, удовлетворяющих F или E высока, то вероятно это точки водяных знаков, поэтому такие кадры мы выбрасываем

Выбор следующего лучшего кадра



Source: Structure-From-Motion Revisited. CVPR2016

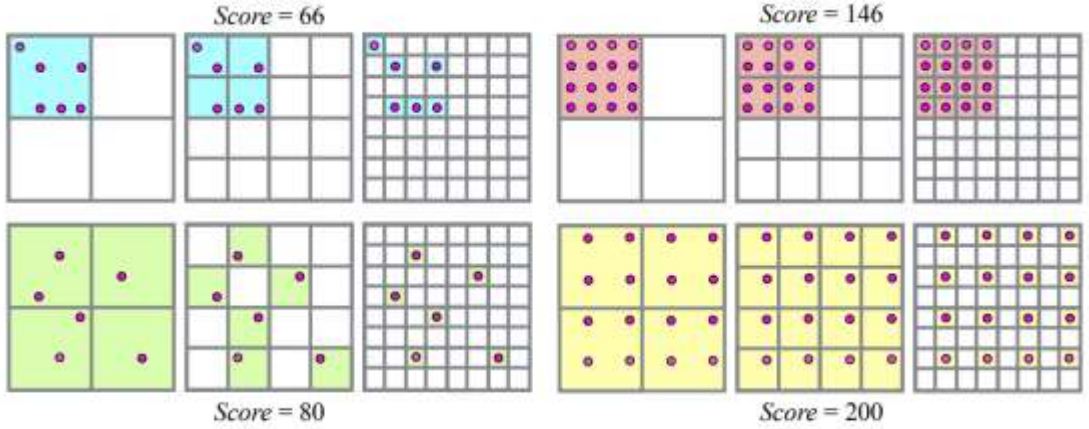
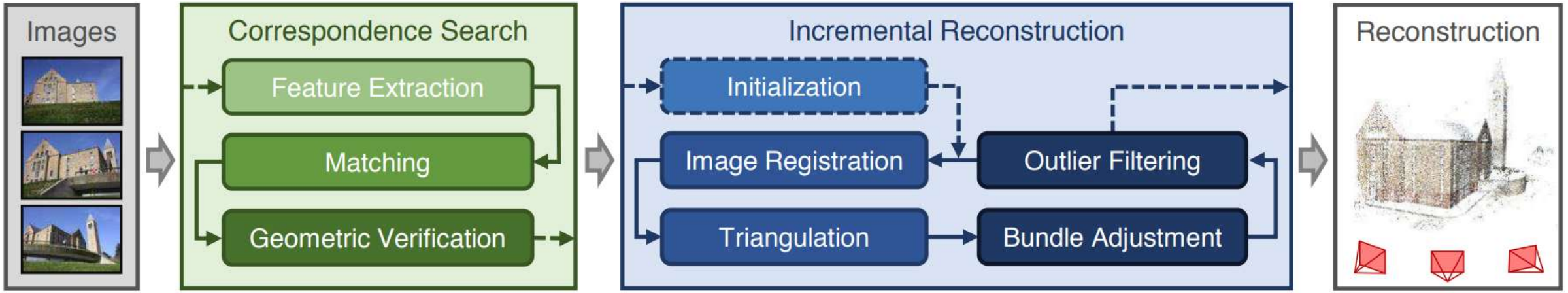


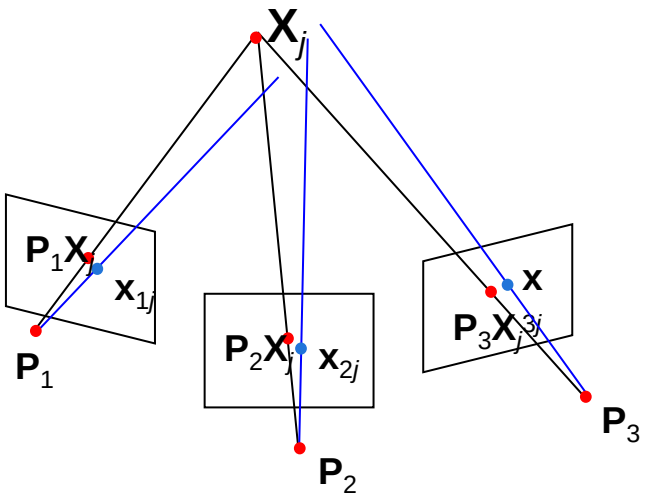
Figure 3. Scores for different number of points (left and right) with different distributions (top and bottom) in the image for $L = 3$.

- Выбор следующего кадра для итеративного метода особенно важен, т.к. ошибки могут накапливаться
- Рассматриваем кадры, у которых $> N_T$ триангулированных точек
- Предложена процедура оценки качества распределения точек
- Разбиваем изображение сеткой, считаем число заполненных клеток, каждой назначая вес пропорционально площади

Робастная триангуляция

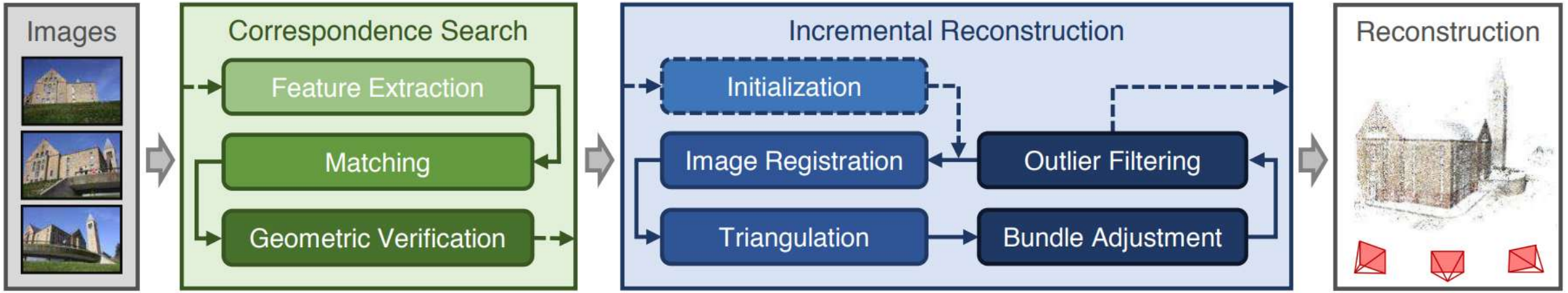


Source: Structure-From-Motion Revisited. CVPR2016

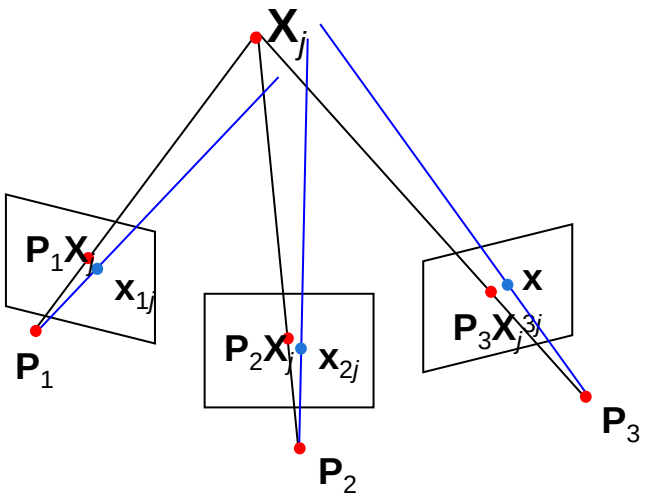


- Предполагаем, что в следах могут быть ошибки, поэтому применяем метод RANSAC для многокадровой триангуляции
- Сэмлпируем только уникальные пары точек
- Изначально предполагаем малую долю истинных соответствий, пересчитываем оценку в процессе RANSAC

Bundle Adjustment

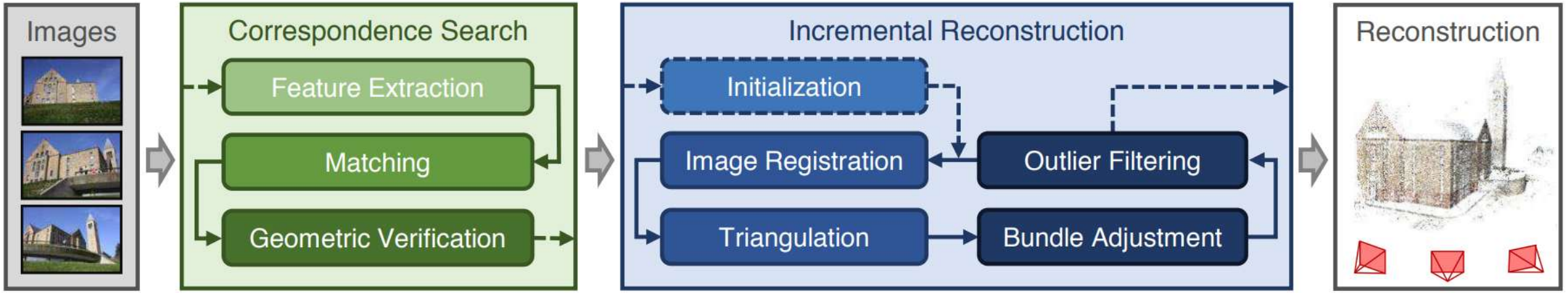


Source: Structure-From-Motion Revisited. CVPR2016

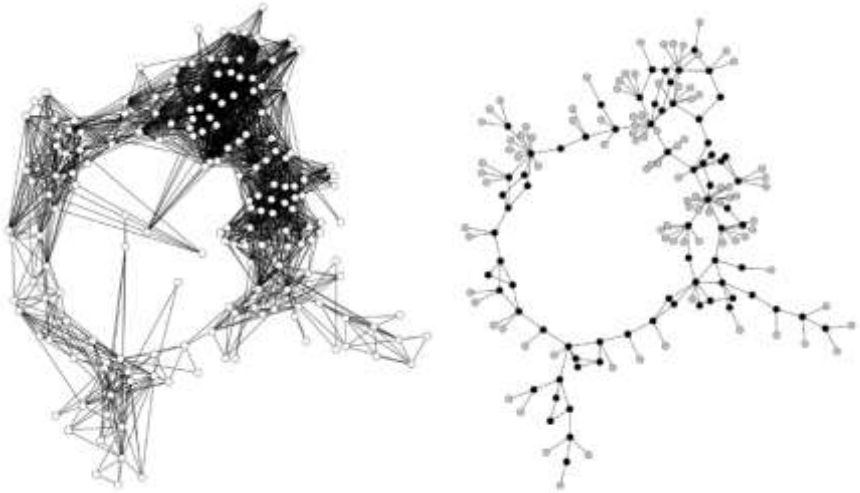


- Применяем Local BA на маленьких группах тесно связанных кадров после добавления каждого кадра
- Применяем Global BA только при увеличении 3Д модели на определённую величину
- Итеративная схема с фильтрацией:
  - Перед BA повторно триангулируем точки и фильтруем ложные (чтобы не портили BA)
  - После BA снова триангулируем, чтобы дополнить модель точками, которые могли плохо оцениваться из-за ошибках в позах и т.д., и отфильтровать ложные.
  - Повторяем BA, поскольку в прошлую итерацию могли использоваться ложные и искажать всё

Redundant View Mining



Source: Structure-From-Motion Revisited. CVPR2016



- Можем «схлопывать» очень близкие ракурсы в один
- Это существенно ускоряет работу, в первую очередь BA без потери точности

Сравнение с аналогами

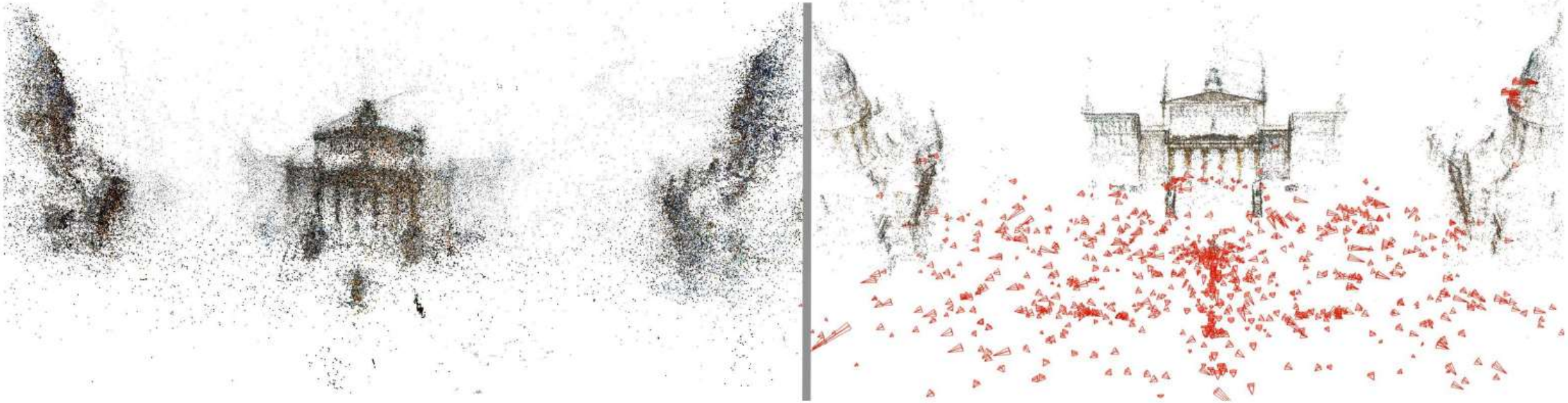


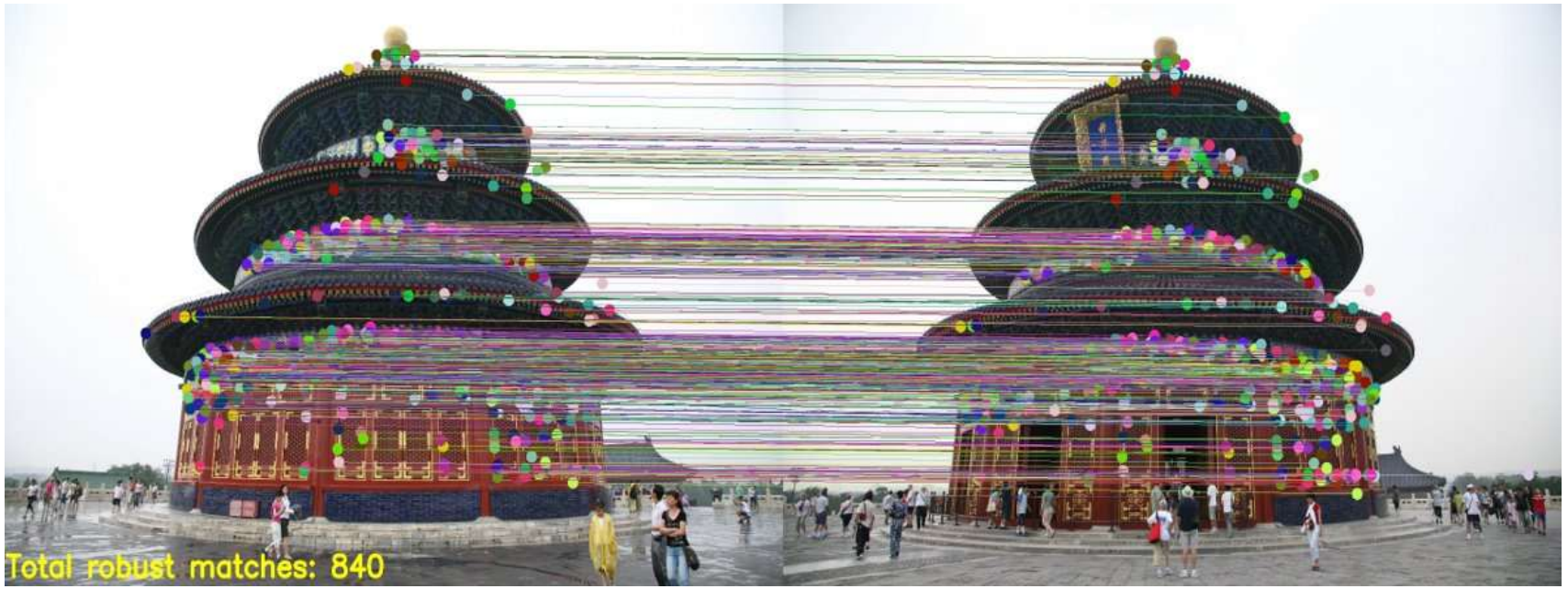
Figure 9. Reconstruction of Gendarmenmarkt [61] for *Bundler* (left) and our method (right).

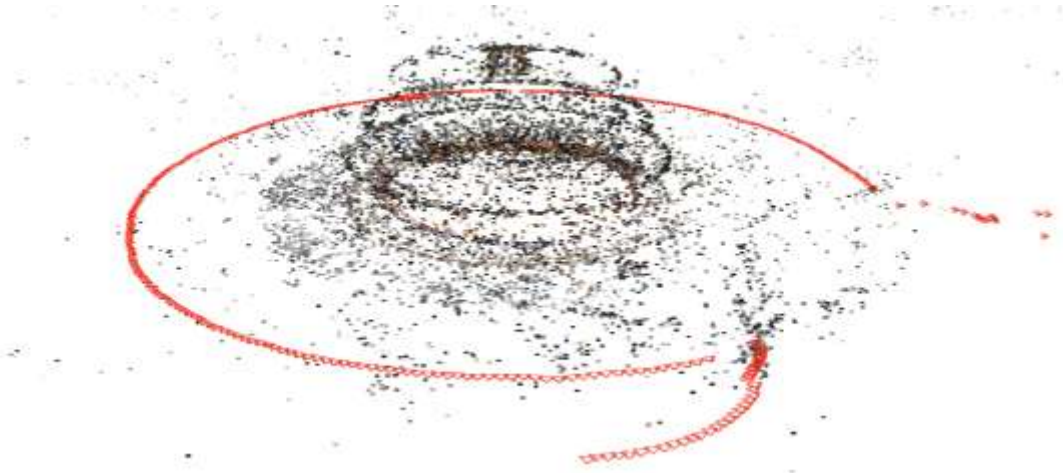
Сравнение с аналогами

	# Images	# Registered				# Points (Avg. Track Length)				Time [s]				Avg. Reproj. Error [px]			
		Theia	*Bundler*	*VSFM*	*Ours*	*Theia*	*Bundler*	*VSFM*	*Ours*	*Theia*	*Bundler*	*VSFM*	*Ours*	*Theia*	*Bundler*	*VSFM*	*Ours*
Rome [14]	74,394	–	13,455	14,797	20,918	–	5.4M	12.9M	5.3M	–	295,200	6,012	10,912	–	–	–	–
Quad [14]	6,514	–	5,028	5,624	5,860	–	10.5M	0.8M	1.2M	–	223,200	2,124	3,791	–	–	–	–
Dubrovnik [36]	6,044	–	–	–	5,913	–	–	–	1.35M	–	–	–	3,821	–	–	–	–
Alamo [61]	2,915	582	647	609	666	146K (6.0)	127K (4.5)	124K (8.9)	94K (11.6)	874	22,025	495	882	1.47	2.29	0.70	0.68
Ellis Island [61]	2,587	231	286	297	315	29K (4.9)	39K (4.1)	61K (5.5)	64K (6.8)	94	12,798	240	332	2.41	2.24	0.71	0.70
Gendarmenmarkt [61]	1,463	703	302	807	861	87K (3.8)	93K (3.7)	138K (4.9)	123K (6.1)	202	465,213	412	627	2.19	1.59	0.71	0.68
Madrid Metropolis [61]	1,344	351	330	309	368	47K (5.0)	27K (3.2)	48K (5.2)	43K (6.6)	95	21,633	203	251	1.48	1.62	0.59	0.60
Montreal Notre Dame [61]	2,298	464	501	491	506	154K (5.4)	135K (4.6)	110K (7.1)	98K (8.7)	207	112,171	418	723	2.01	1.92	0.88	0.81
NYC Library [61]	2,550	339	400	411	453	66K (4.1)	71K (3.7)	95K (5.5)	77K (7.1)	194	36,462	327	420	1.89	1.84	0.67	0.69
Piazza del Popolo [61]	2,251	335	376	403	437	36K (5.2)	34K (3.7)	50K (7.2)	47K (8.8)	89	33,805	275	380	2.11	1.76	0.76	0.72
Piccadilly [61]	7,351	2,270	1,087	2,161	2,336	197K (4.9)	197K (3.9)	245K (6.9)	260K (7.9)	1,427	478,956	1,236	1,961	2.33	1.79	0.79	0.75
Roman Forum [61]	2,364	1,074	885	1,320	1,409	261K (4.9)	281K (4.4)	278K (5.7)	222K (7.8)	1,302	587,451	748	1,041	2.07	1.66	0.69	0.70
Tower of London [61]	1,576	468	569	547	578	140K (5.2)	151K (4.8)	143K (5.7)	109K (7.4)	201	184,905	497	678	1.86	1.54	0.59	0.61
Trafalgar [61]	15,685	5,067	1,257	5,087	5,211	381K (4.8)	196K (3.7)	497K (8.7)	450K (10.1)	1,494	612,452	3,921	5,122	2.09	2.07	0.79	0.74
Union Square [61]	5,961	720	649	658	763	35K (5.3)	48K (3.7)	43K (7.1)	53K (8.2)	131	56,317	556	693	2.36	3.22	0.68	0.67
Vienna Cathedral [61]	6,288	858	853	890	933	259K (4.9)	276K (4.6)	231K (7.6)	190K (9.8)	764	567,213	899	1,244	2.45	1.69	0.80	0.74
Yorkminster [61]	3,368	429	379	427	456	143K (4.5)	71K (3.9)	130K (5.2)	105K (6.8)	164	34,641	661	997	2.38	2.61	0.72	0.70

Table 1. Reconstruction results for state-of-the-art SfM systems on large-scale unordered Internet photo collections.

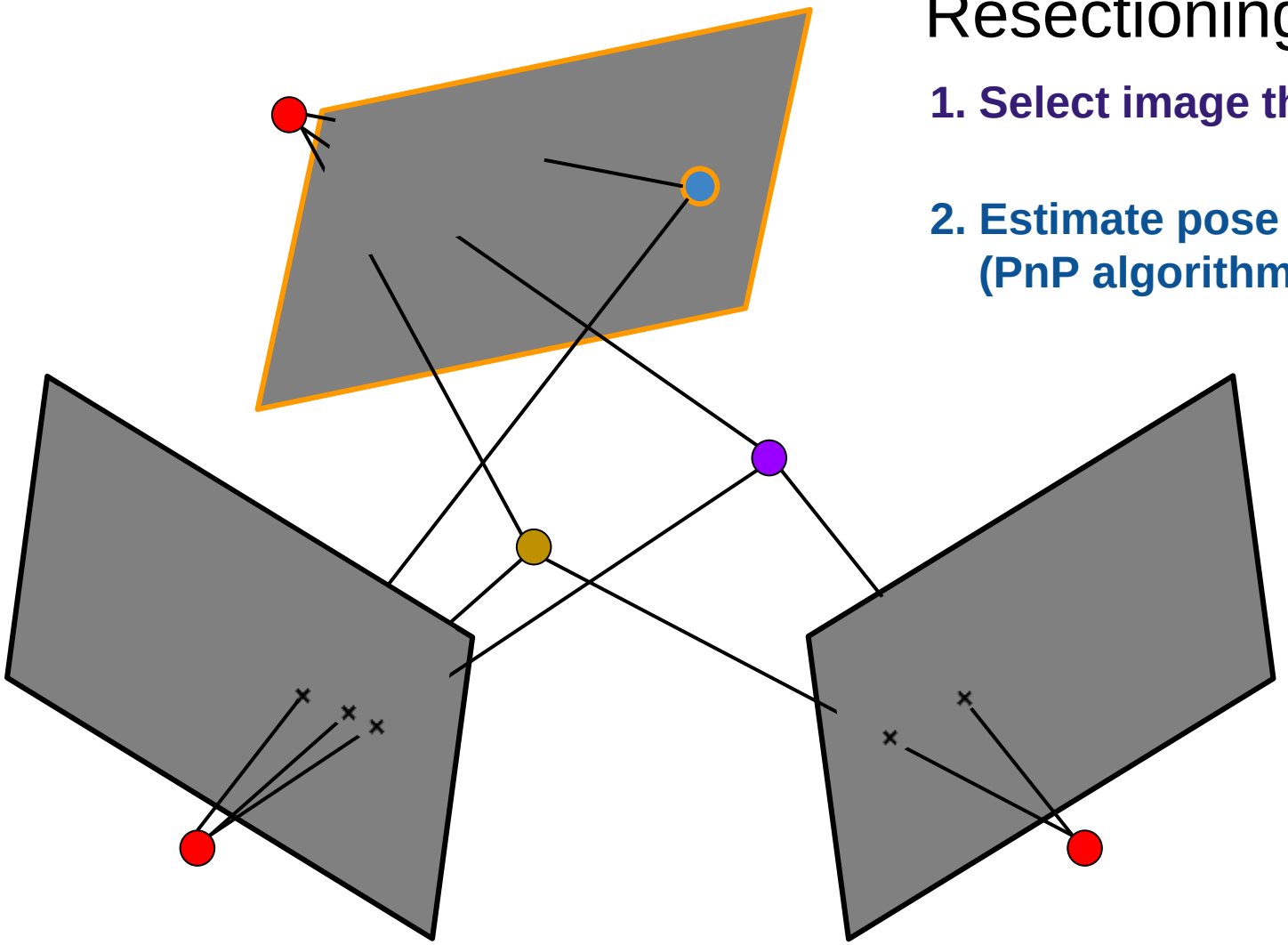
Проблема повторяющихся структур





https://github.com/rajkataria/ReliableResectioning (3DV 2020)

Resectioning – добавление нового кадра

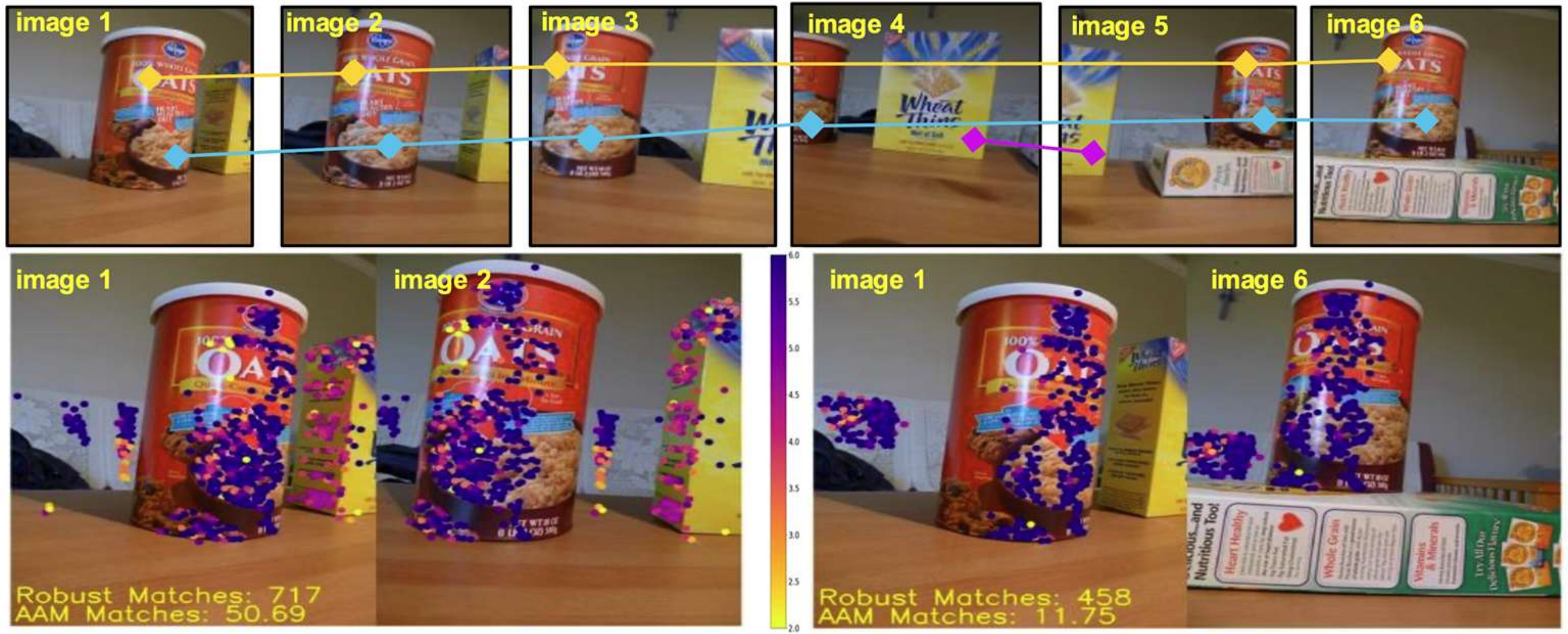


Resectioning is a critical step

1. **Select image that views the most triangulated points**

2. **Estimate pose of image using all the triangulated points (PnP algorithm using RANSAC)**

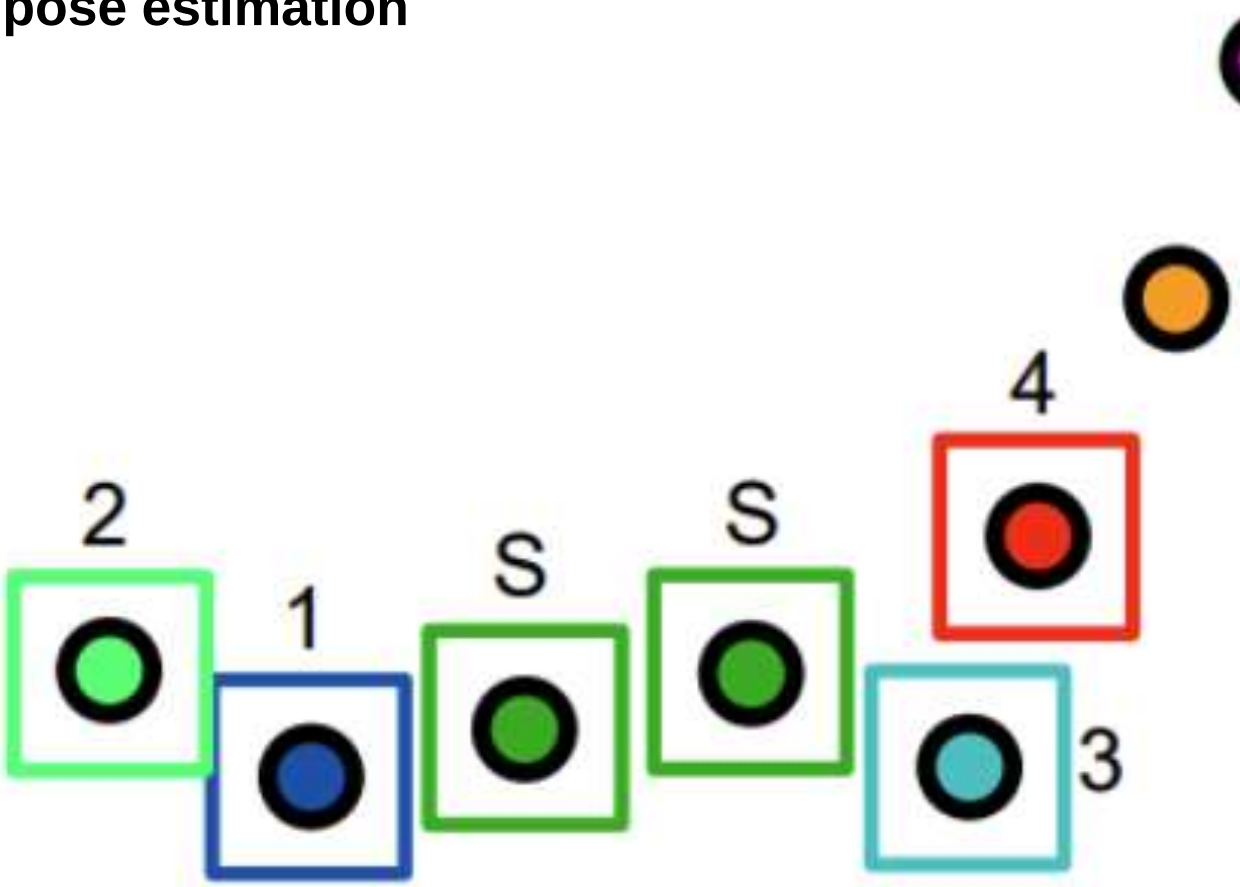
Пенализация длинных следов



Ambiguity-adjusted match score (AAM): Discount longer tracks that are more likely to correspond to duplicate structures

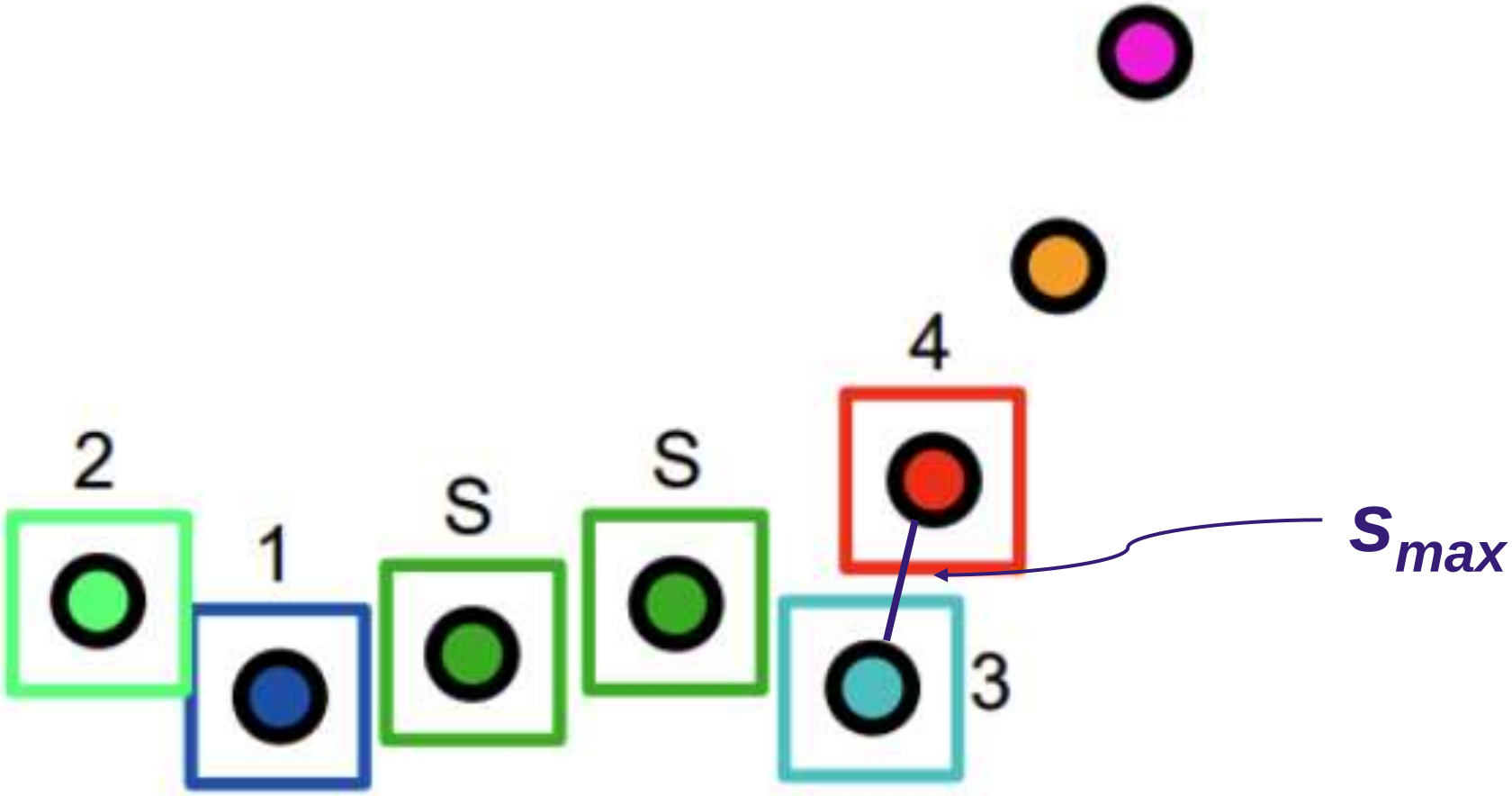
Сортировка кадров по AAM score

- Local resectioning order uses most similar image
- We use points from a smaller set of reliable images to determine **resectioning order** and **pose estimation**



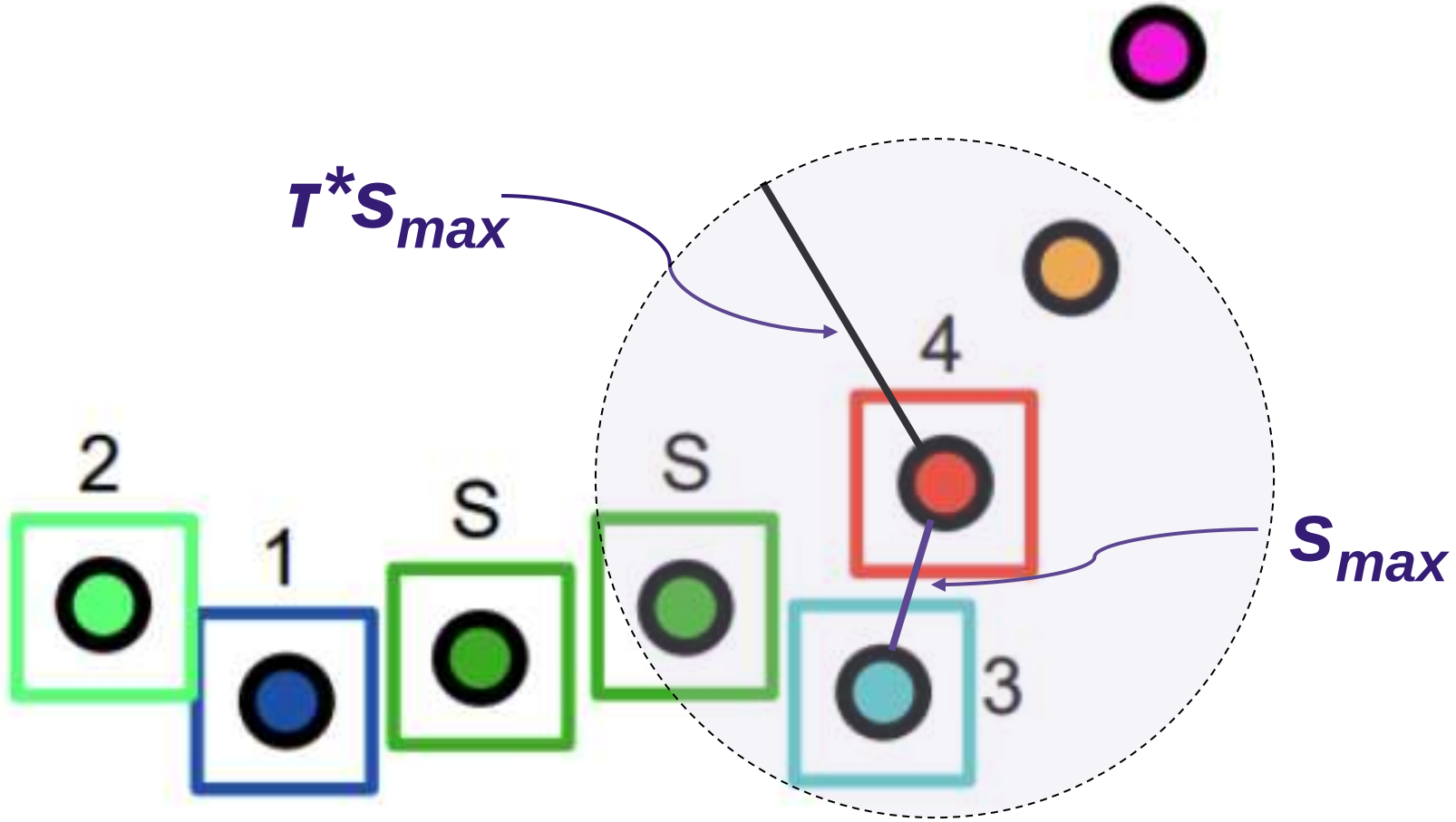
Подход к ресекции

- Local pose estimation uses reliable images only
- Use points from a smaller set of reliable images to determine **resectioning order** and **pose estimation**



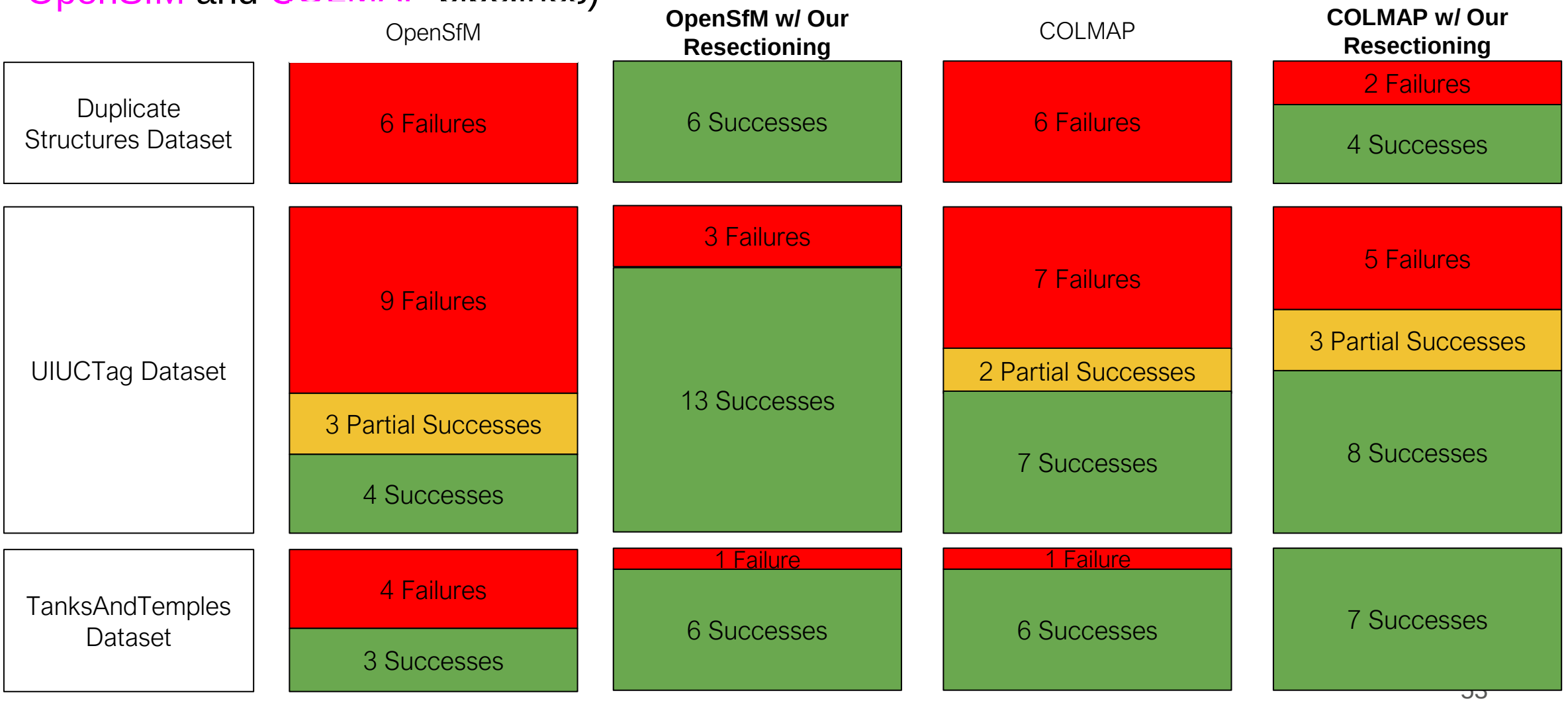
Подход к ресекции

- Local pose estimation uses reliable images only
- Use points from a smaller set of reliable images to determine **resectioning order** and **pose estimation**



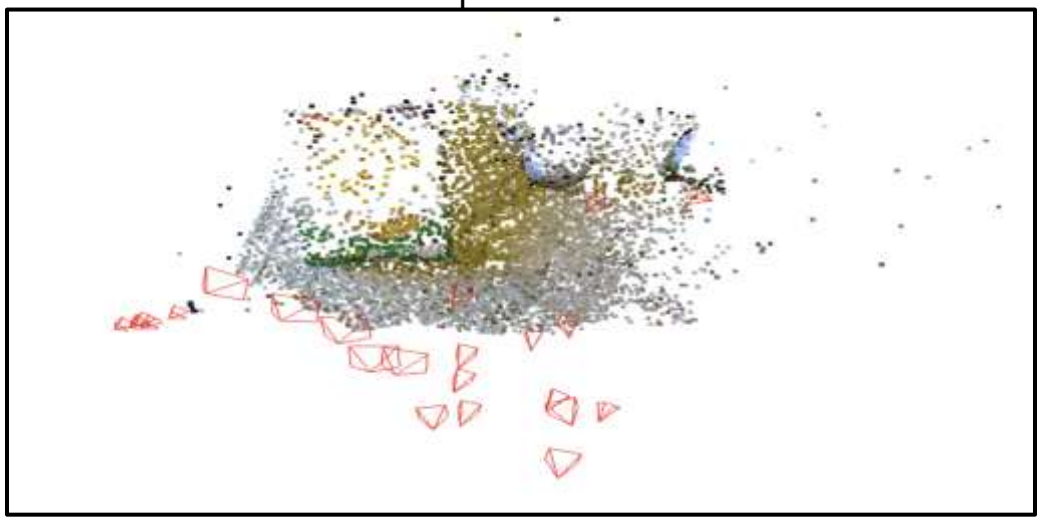
Улучшение стандартных подходов

Local resectioning using ambiguity-adjusted matches compared against baselines (standard OpenSfM and COLMAP pipelines)

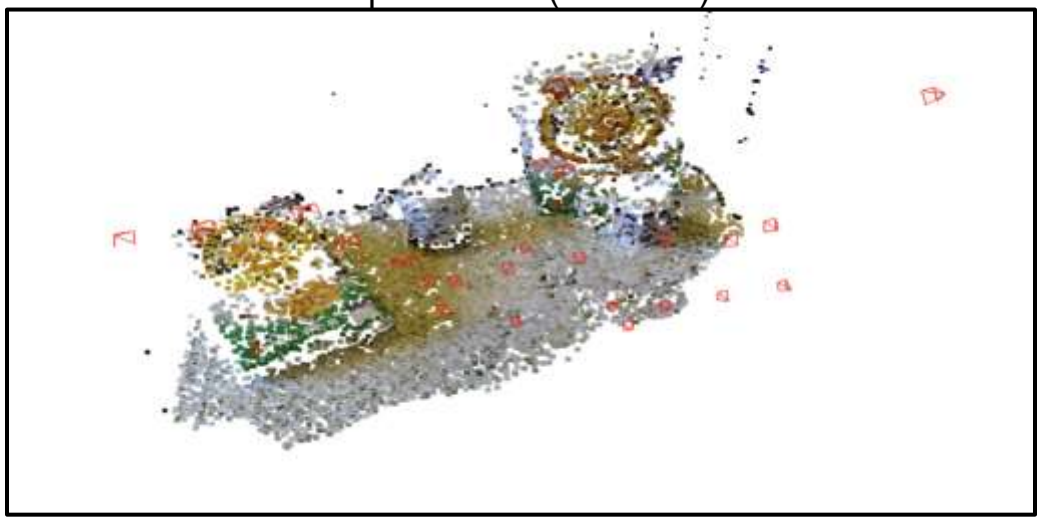


Пример - Cereal (DuplicateStructures)

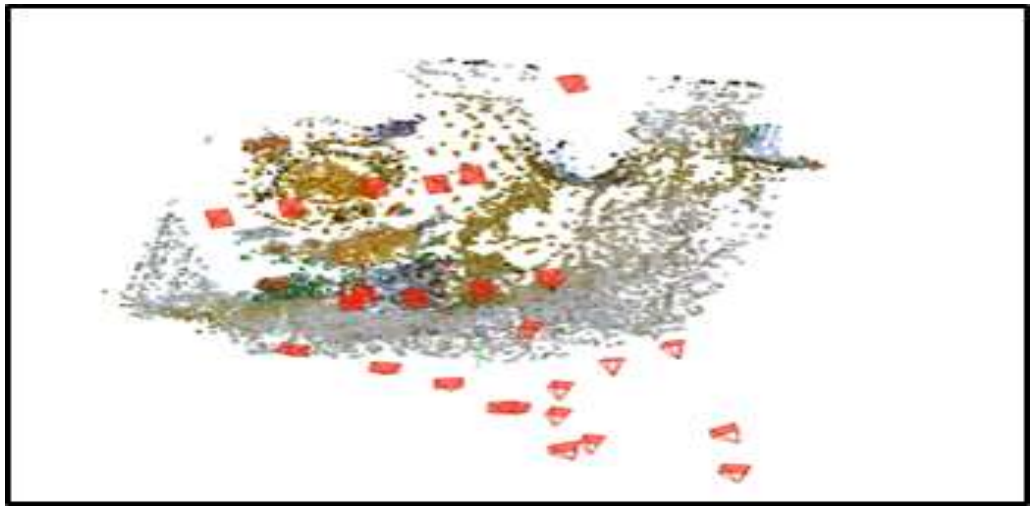
OpenSfM



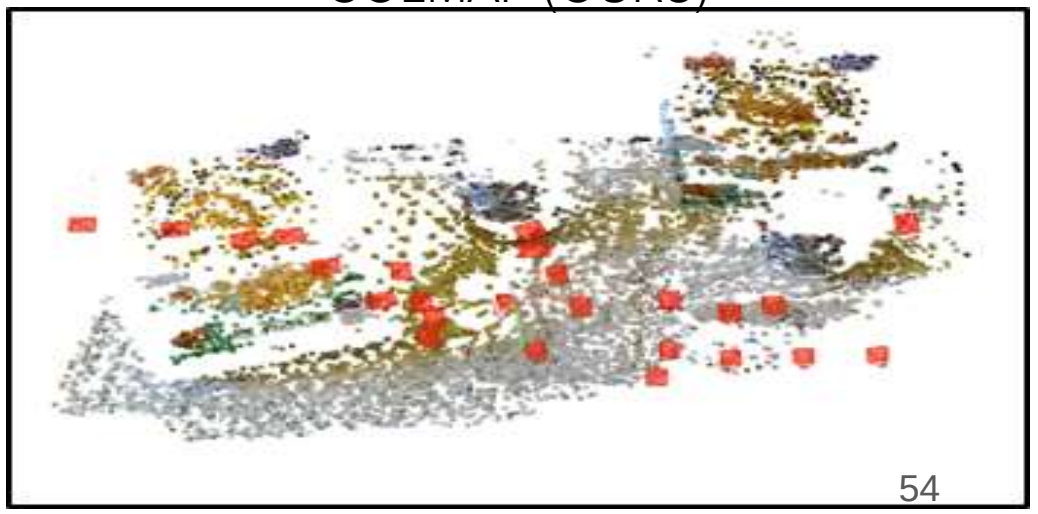
OpenSfM (OURS)



COLMAP

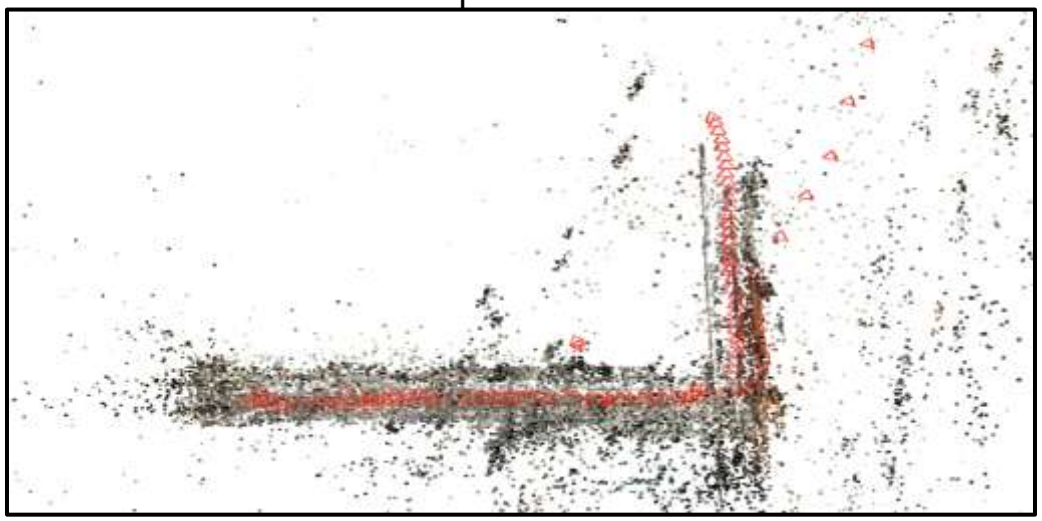


COLMAP (OURS)

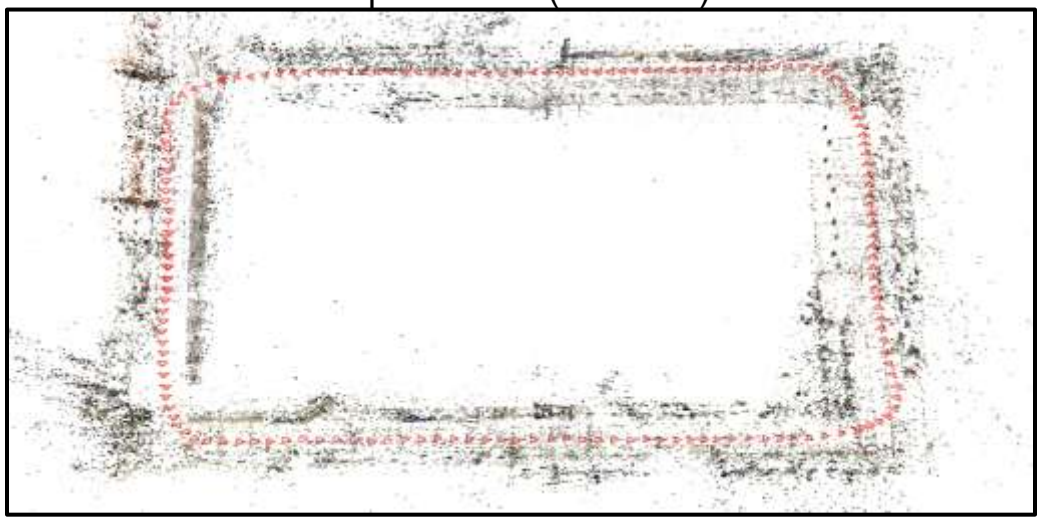


Пример - ece_floor3_loop_cw (UIUCTag)

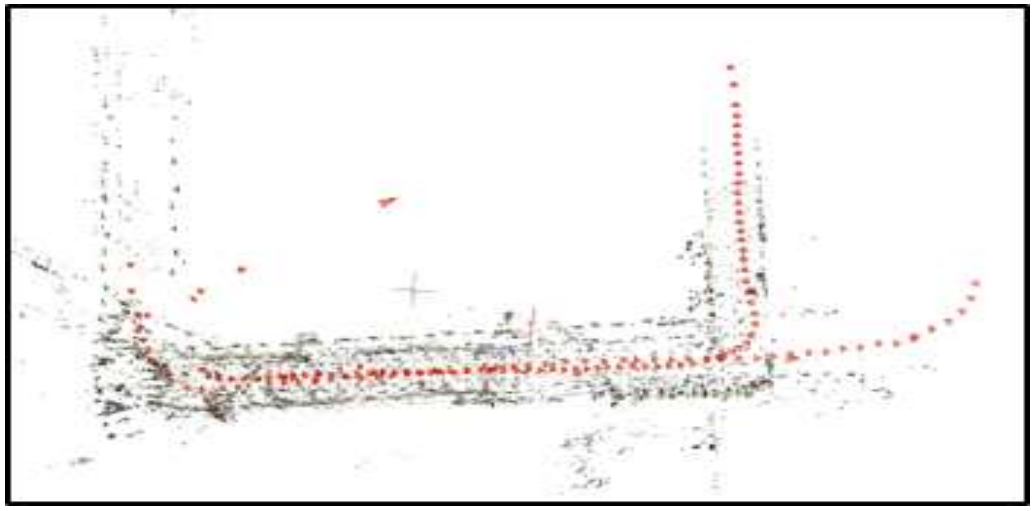
OpenSfM



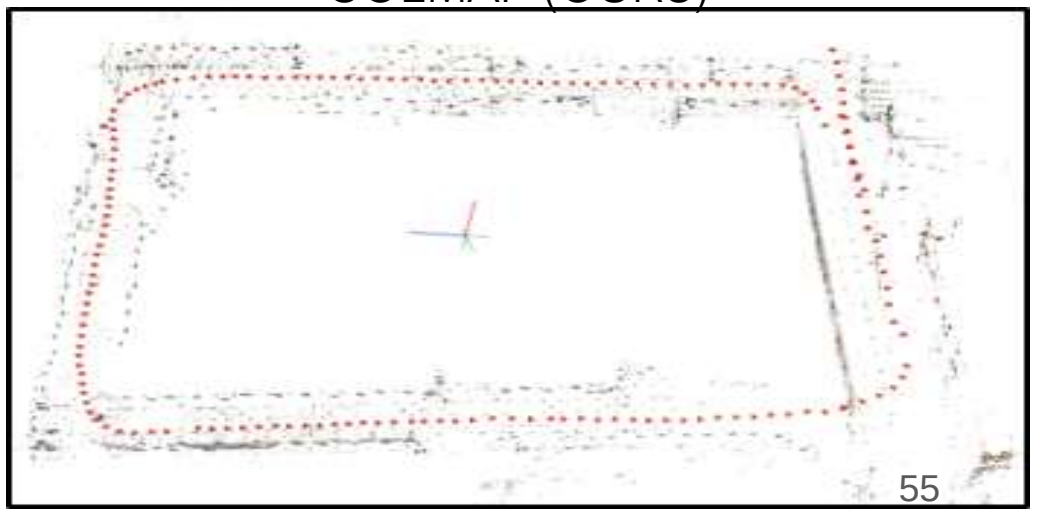
OpenSfM (OURS)



COLMAP

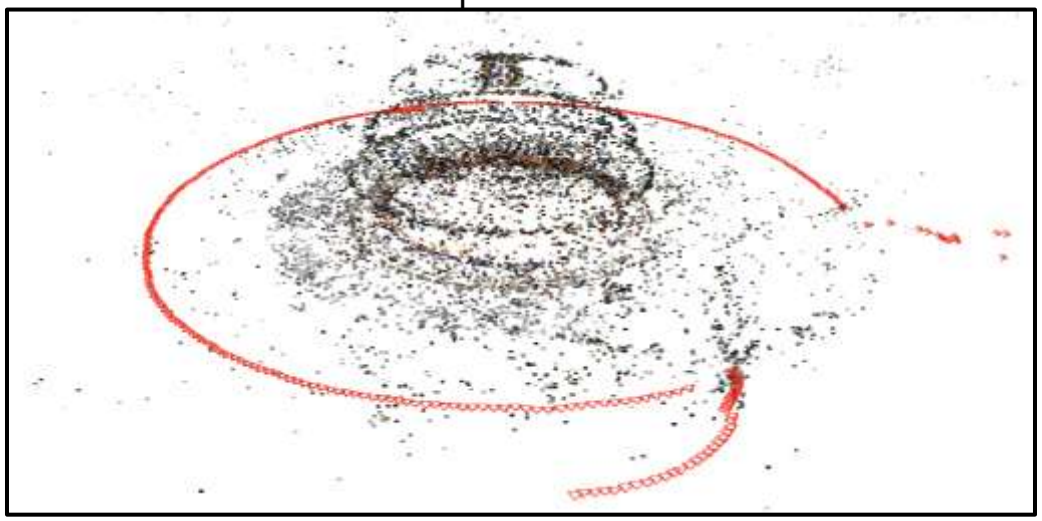


COLMAP (OURS)

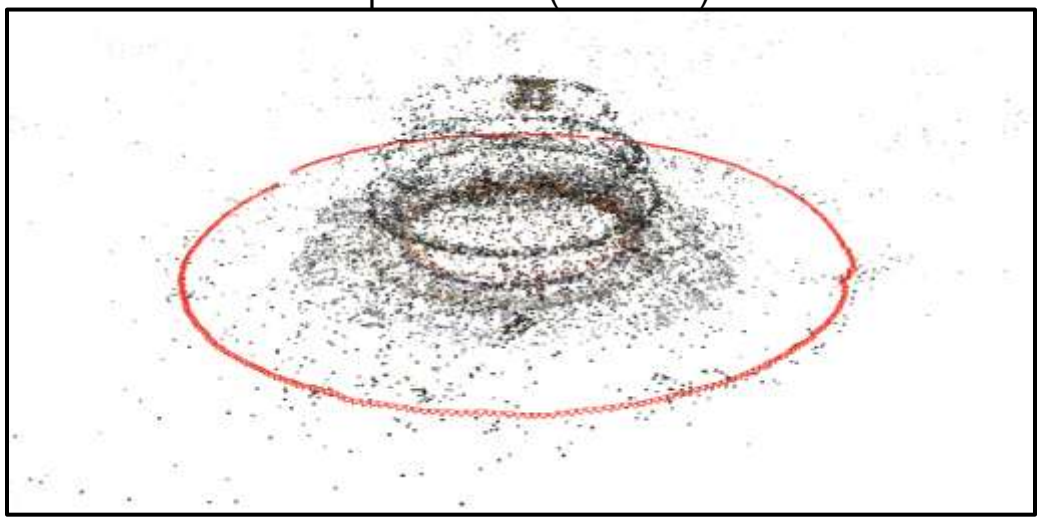


Пример - TempleOfHeaven (Internet)

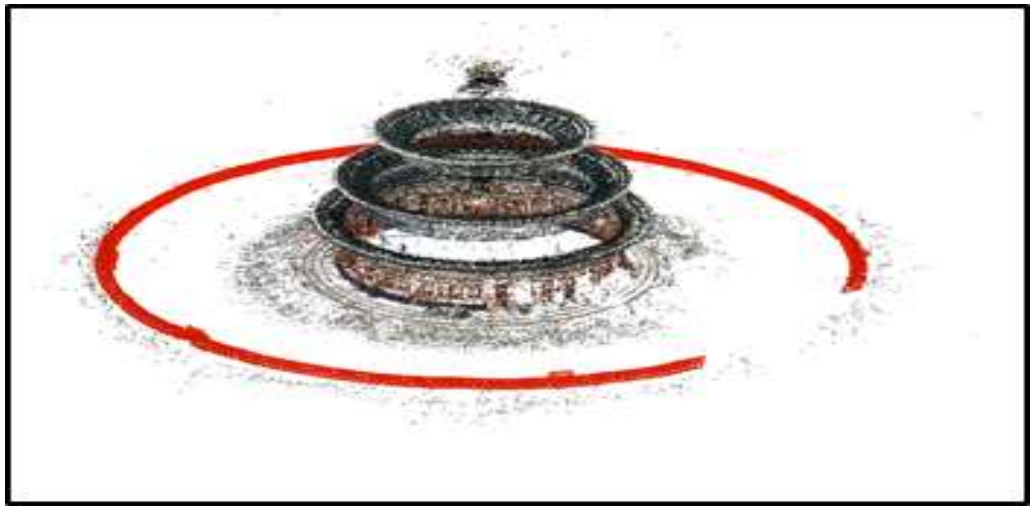
OpenSfM



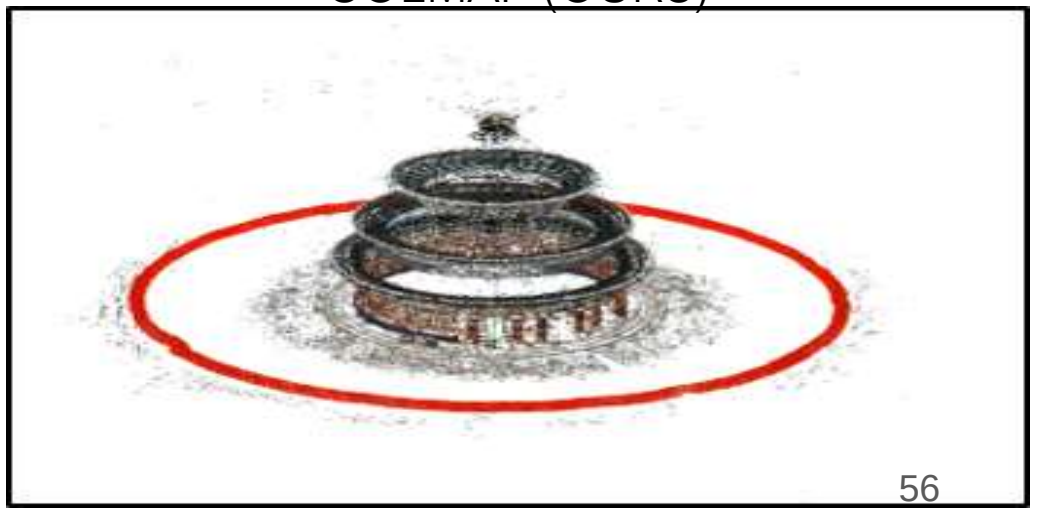
OpenSfM (OURS)



COLMAP



COLMAP (OURS)



Можно ли улучшить за счёт DL-точек?

2021

Image Matching Across Wide Baselines: From Paper to Practice

Yuhe Jin · Dmytro Mishkin · Anastasiia Mishchuk · Jiri Matas · Pascal Fua · Kwang Moo Yi · Eduard Trulls

Local featured type	Number of Images				
	100	200	400	800	all
SIFT [54]	0.06°	0.09°	0.06°	0.07°	0.09°
SIFT (Upright) [54]	0.07°	0.07°	0.04°	0.06°	0.09°
HardNet (Upright) [62]	0.06°	0.06°	0.06°	0.04°	0.05°
SuperPoint [34]	0.31°	0.25°	0.33°	0.19°	0.32°
R2D2 [80]	0.12°	0.08°	0.07°	0.08°	0.05°

Table 2 Standard deviation of the pose difference of three COLMAP runs with different number of images. Most of them are below 0.1°, except for SuperPoint.

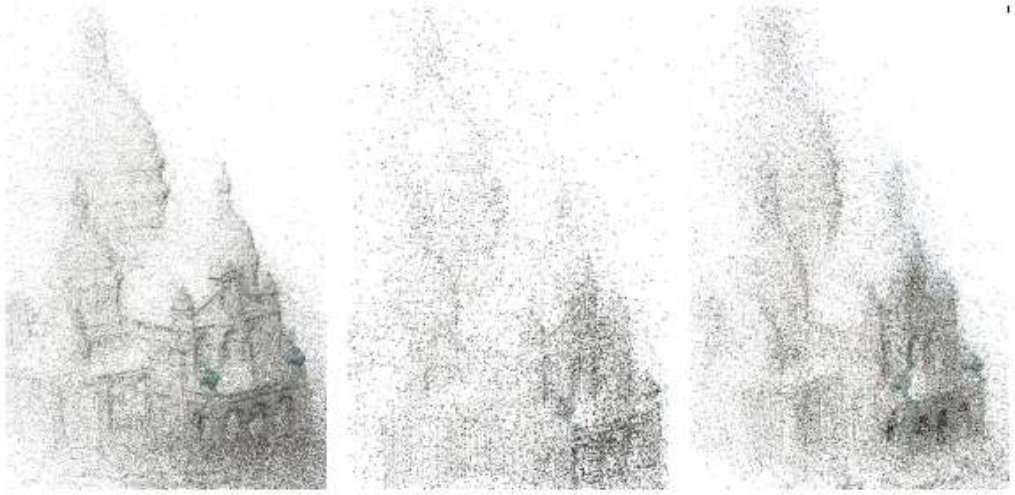
Local feature type	Number of images			
	100 vs. all	200 vs. all	400 vs. all	800 vs. all
SIFT [54]	0.58° / 0.22°	0.31° / 0.08°	0.23° / 0.05°	0.18° / 0.04°
SIFT (Upright) [54]	0.52° / 0.16°	0.29° / 0.08°	0.22° / 0.05°	0.16° / 0.03°
HardNet (Upright) [62]	0.35° / 0.10°	0.33° / 0.08°	0.23° / 0.06°	0.14° / 0.04°
SuperPoint [34]	1.22° / 0.71°	1.11° / 0.67°	1.08° / 0.48°	0.74° / 0.38°
R2D2 [80]	0.49° / 0.14°	0.32° / 0.10°	0.25° / 0.08°	0.18° / 0.05°

Table 3 Pose convergence in SfM. We report the mean/median of the difference (in degrees) between the poses extracted with the full set of 1179 images for "Sacre Coeur", and different subsets of it, for four local feature methods – to keep the results comparable we only look at the 100 images in common across all subsets. We report the maximum among the angular difference between rotation matrices and translation vectors. The estimated poses are stable, with as little as 100 images.

Можно ли улучшить за счёт DL-точек?

Reference	Compared to			
	SIFT (Upright)	HardNet (Upright)	SuperPoint	R2D2
SIFT [54]	0.20° / 0.05°	0.26° / 0.05°	1.01° / 0.62°	0.26° / 0.09°

Table 4 Difference between poses obtained with different local features. We report the mean/median of the difference (in degrees) between the poses extracted with SIFT (Upright), HardNet (Upright), SuperPoint, or R2D2, and those extracted with SIFT. We use the maximum of the angular difference between rotation matrices and translation vectors. SIFT (Upright), HardNet (Upright), and R2D2 give near-identical results to SIFT.



(a) SIFT (b) SuperPoint (c) R2D2

Fig. 6 COLMAP with different local features. We show the reconstructed point cloud for the scene "Sacre Coeur" using three different local features: SIFT, SuperPoint, and R2D2, using all images available (1179). The reconstructions with SIFT and R2D2 are both dense, albeit somewhat different. The reconstruction with SuperPoint is quite dense, considering it can only extract a much smaller number of features effectively, but its poses appear less accurate.

Резюме SfM



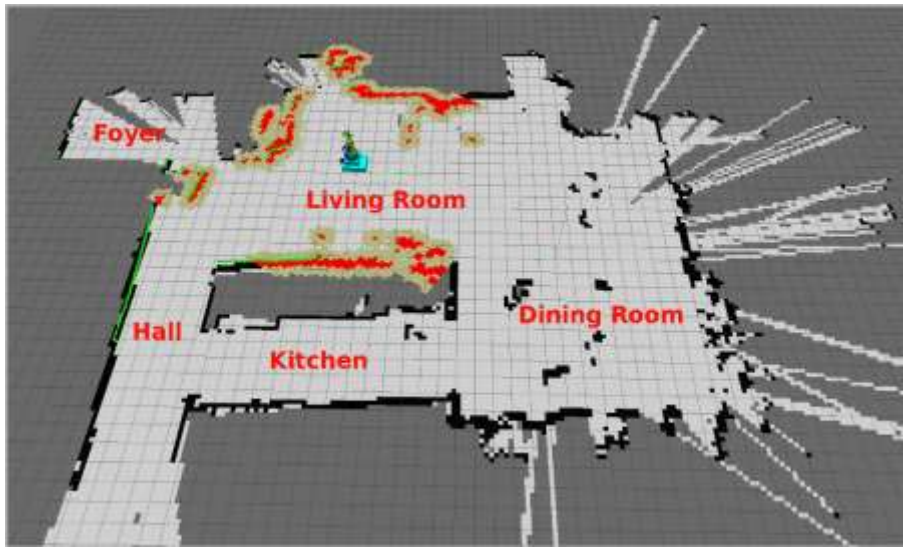
- Разреженная трёхмерная реконструкция – один из наиболее проработанных разделов компьютерного зрения
- Используется для фотограмметрии или как первый этап для последующей плотной 3Д реконструкции или распознавания
- Есть хорошие открытые библиотеки
- Для решения отдельных подзадач сейчас исследуют нейросетевые методы
- Задачу нельзя считать полностью решённой, не для всех кадров удаётся найти позу, бывают серьёзные сбои, много шумных точек



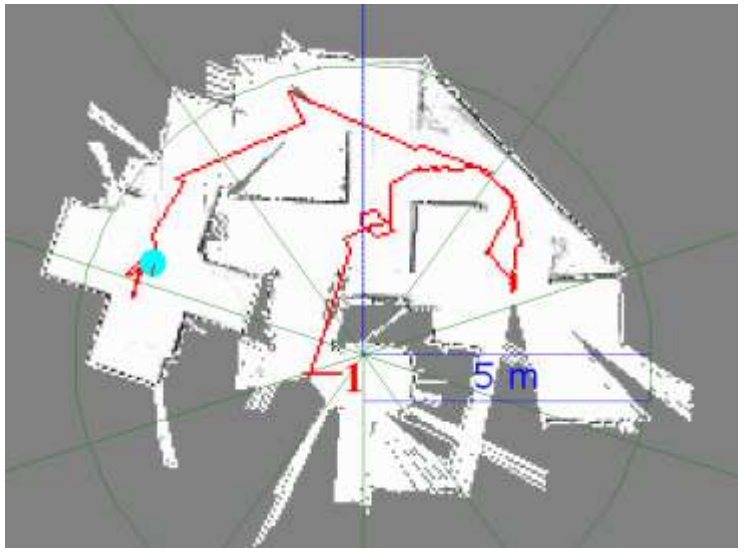
SLAM

Simultaneous Localization and Mapping (SLAM)

Комбинация Одометрии (оценки траектории движения камеры), Картографирования (построения карты сцены), Локализации (оценки положения по карте), Закрытия циклов / Loop Closure



Карта какой-то квартиры



Карта и траектория

Если по данным камеры, тогда говорят Visual SLAM. Но могут использоваться и другие сенсоры (IMU, 2D Lidar, 3D Lidar, и т.д.)

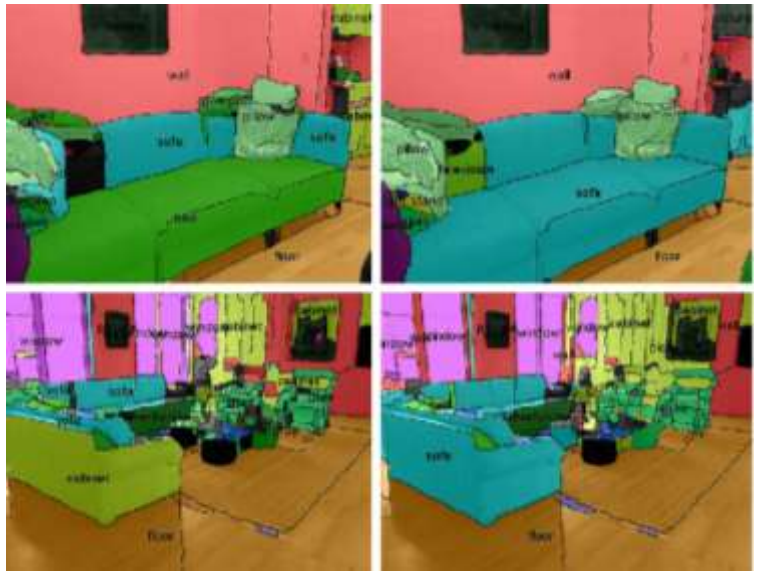
SfM vs. SLAM



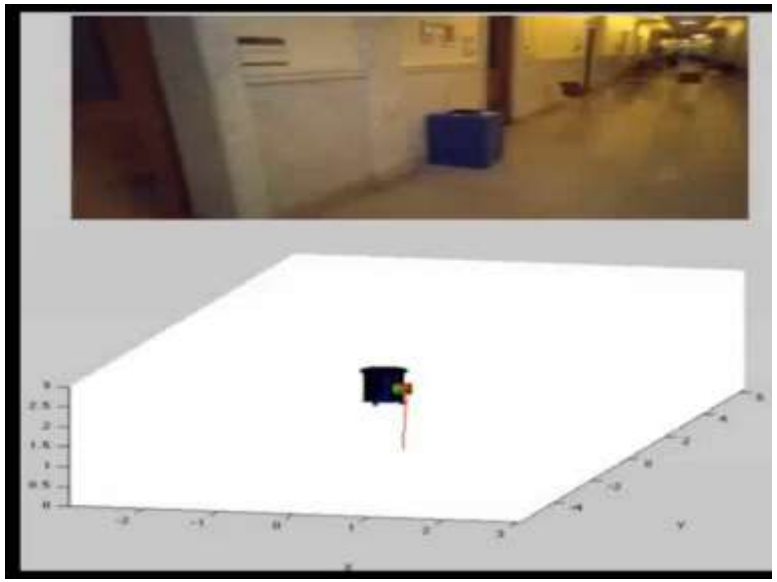
SfM	SLAM
• Input is unordered set of images	• Input is stream of images, stereo, or depth and sometimes IMU
• Focus is on precision, with aim to produce a good 3D model	• Focus is on speed and robustness, with aim to localize camera or robot
• Offline, one-time process	• Online process, possibly with relocalization
• Published mainly in vision conferences	• Published mainly in robotics conferences
• 3 papers with more than 1000 citations	• 8 papers with more than 1000 citations
• Complicated	• Very complicated

Semantic SLAM

- Semantic SLAM или Total Scene Understanding
  - Одновременное распознавание сцены и SLAM, что дает полный ответ на вопрос «что где расположено» про всю сцену

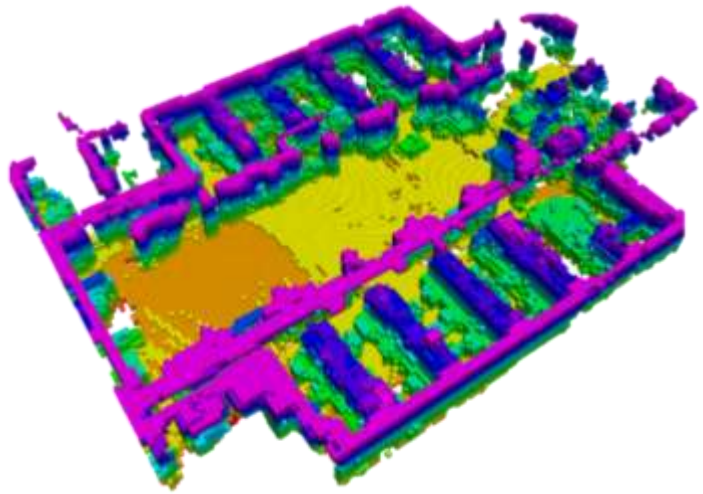


Semantic segmentation



Example of semantic SLAM

Source: https://natanaso.github.io

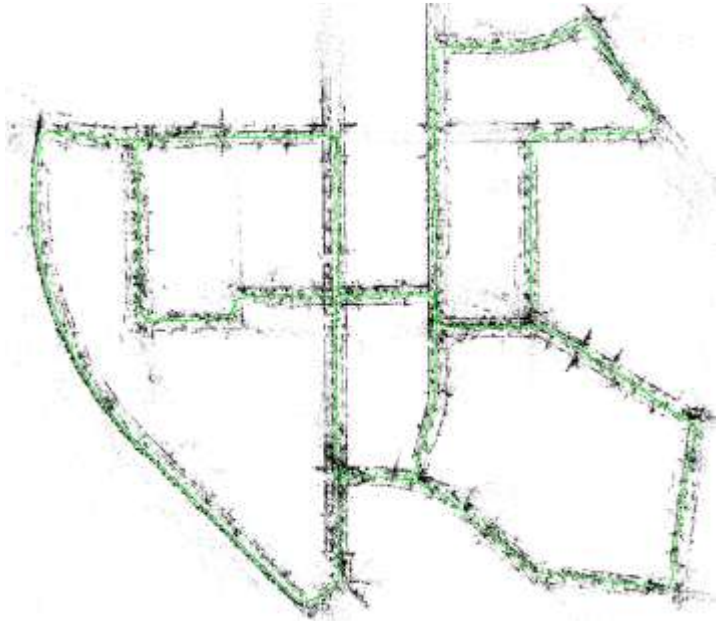


Volumetric semantic map

Source: https://arxiv.org/pdf/1801.07380.pdf

Оценка качества

«Карта» обычно представлена в виде набора ключевых кадров с позами и облаком точек. Непонятно, как оценивать точности таких карт. Поэтому качество SLAM обычно оценивают по траекториям



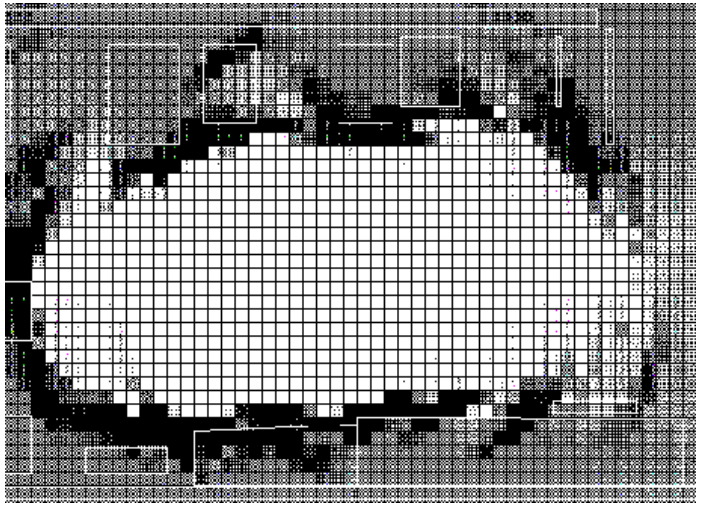
Sparse point cloud from monocular RGB images



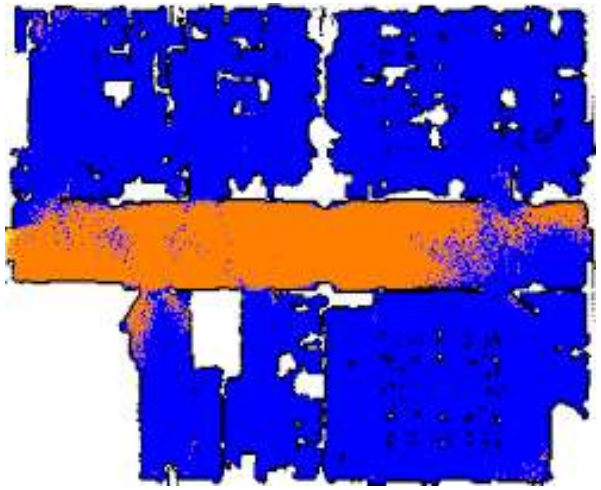
Dense point cloud from RGBD video

Семантические карты

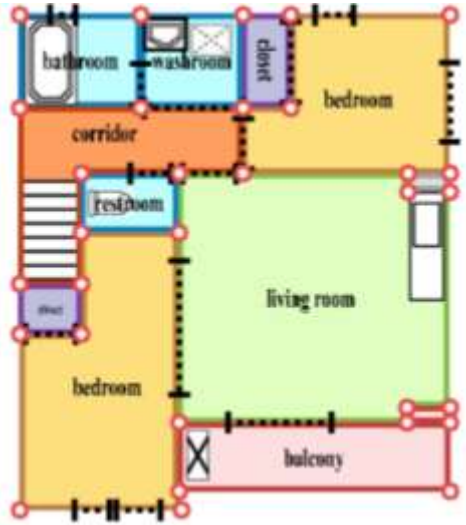
- Если карты строятся, то обычно они строятся в виде 2Д изображений.
- Для оценки точности требуется эталонная разметка этих карт



Occupancy grid map



Occupancy grid map with semantic labels

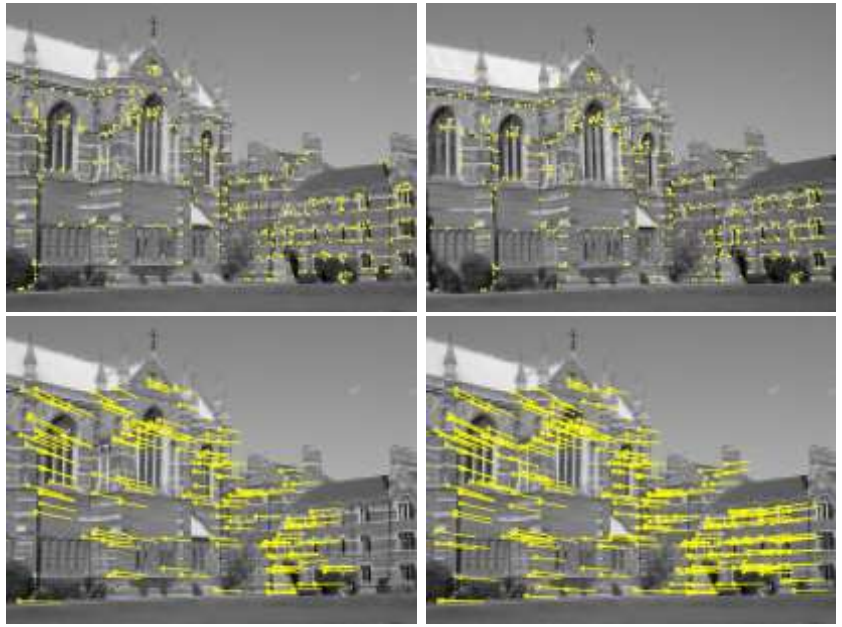


Vector map with labels

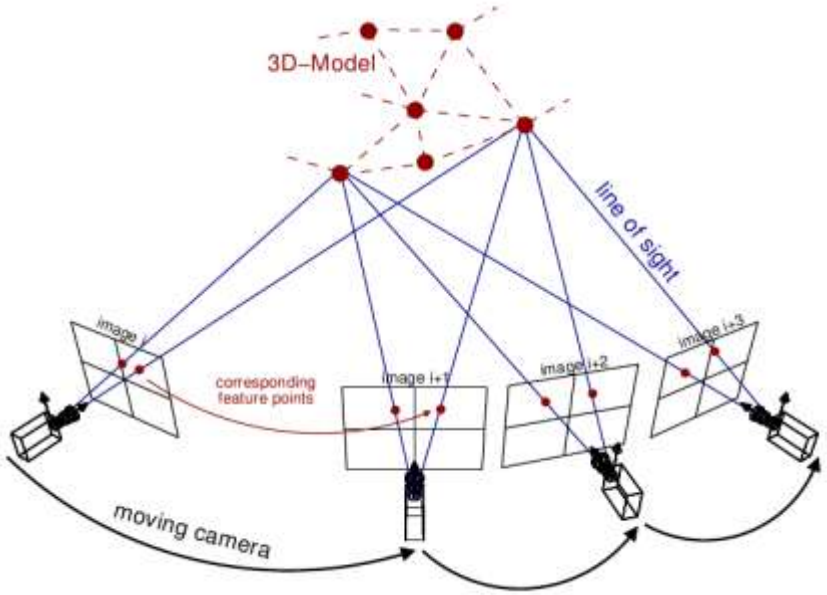
Классический SLAM

Классический Visual SLAM

- Классическими можно назвать методы, опирающиеся на точечные соответствия между изображениями, многовидовую геометрию и уточнение градиентной оптимизацией
- Фокус на последовательную обработку кадров и работу в реальном времени



Точки и их сопоставление



Локализация камеры и структура из движения

ORBSLAM

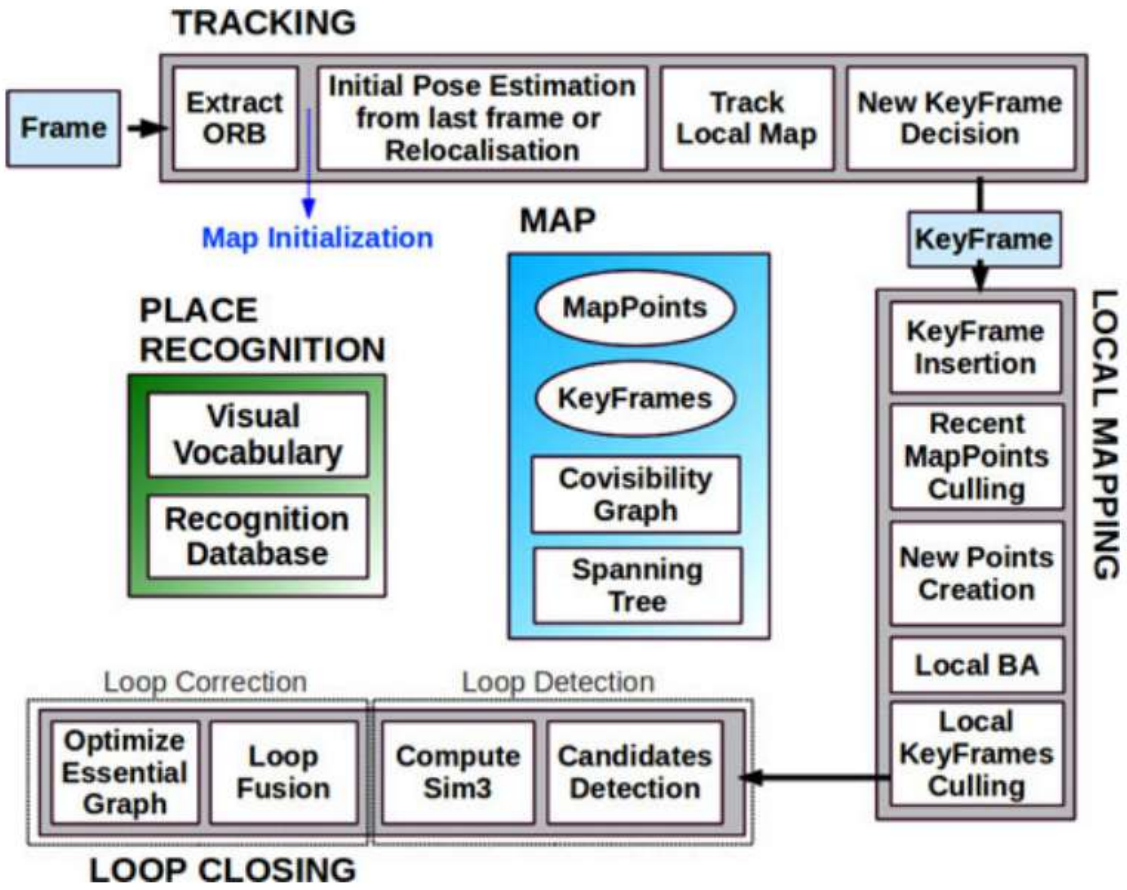


Fig. 1. ORB-SLAM system overview, showing all the steps performed by the tracking, local mapping, and loop closing threads. The main components of the place recognition module and the map are also shown.

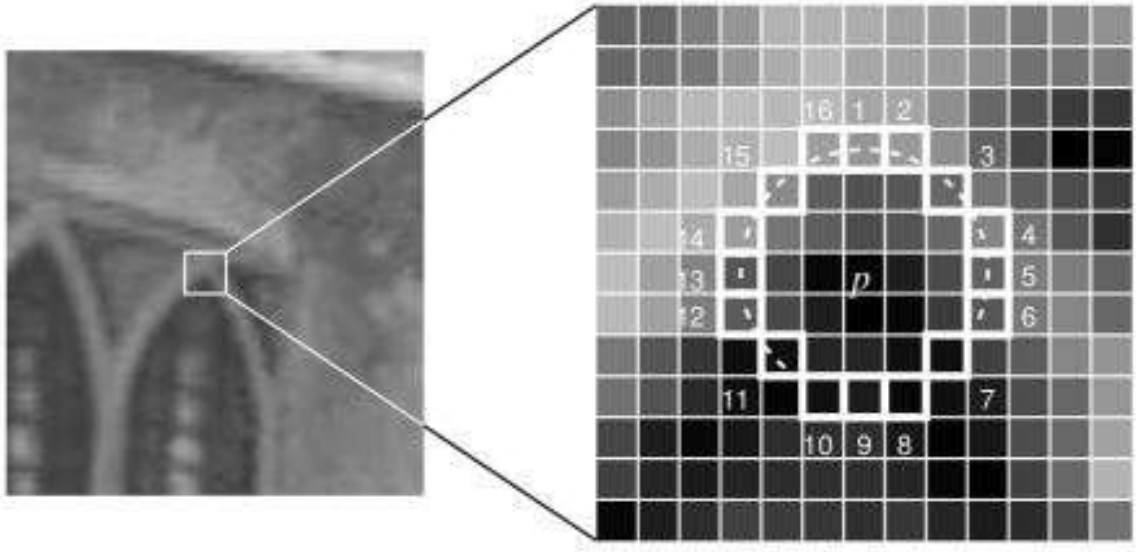
- Первая работа цикла открытых реализацией SLAM-методов
  - https://github.com/raulmur/ORB_SLAM
- Опирается на быстрый детектор ORB
- Карта сцены описывается набором ключевых кадров и набором 3д точек, наблюдаемых на этих кадрах
- 3 нити исполнения (Tracking, Local Mapping, Loop Closing), работающие параллельно
- Tracking – отслеживание точек и оценка позы камеры в локальной карте
- Local Mapping – добавление нового ключевого кадра в карту и последствия этого
- Loop Closure – определение циклов и уточнение карты и траектории из-за этого

R.Mur-Artal, J.M.M.Montiel, J.D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on robotics, 2015

ORB (Oriented FAST and Rotated BRIEF)

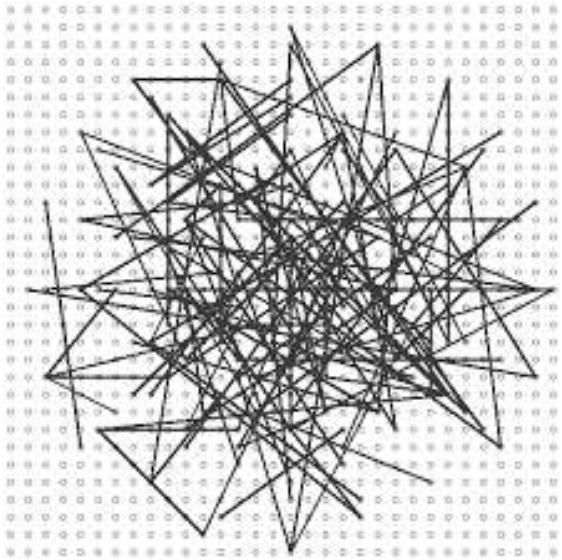
«Вершина» необучаемых аппроксимаций SIFT

Идея FAST



- Выбираем 16 пикселов на дискретной окружности (алгоритм Брезенхейма) вокруг точки p
- Углом будет считаться точка, если 12 последовательных пикселов либо светлее p больше, чем на порог t, либо темнее

Идея BRIEF



- Выбираем N случайных пар пикселов (a_i, b_i) в окрестности
- Каждой паре сопоставляем 1 (если яркость $I(a_i) > I(b_i)$) и 0, если наоборот
- Получаем бинарный код desc(p), и можем сравнить по расстоянию Хэмминга с другими

https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf

ORBSLAM - Tracking

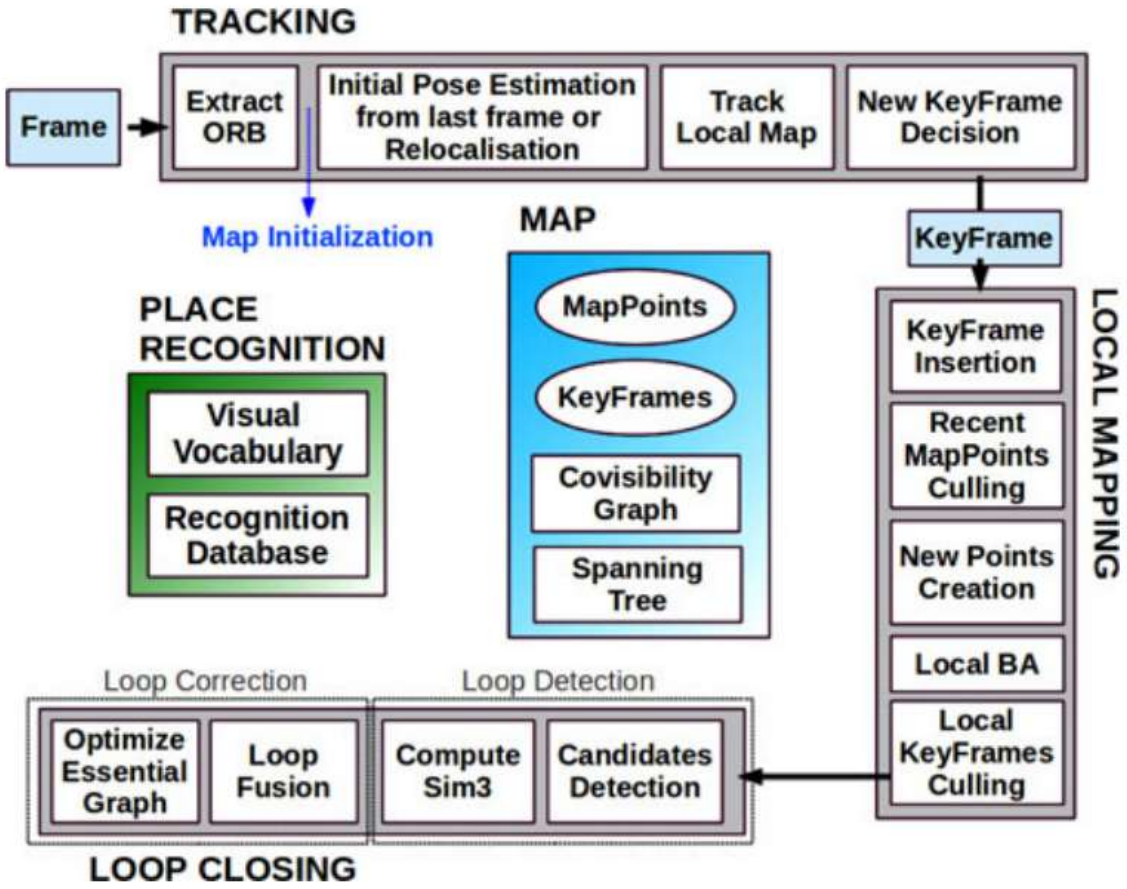
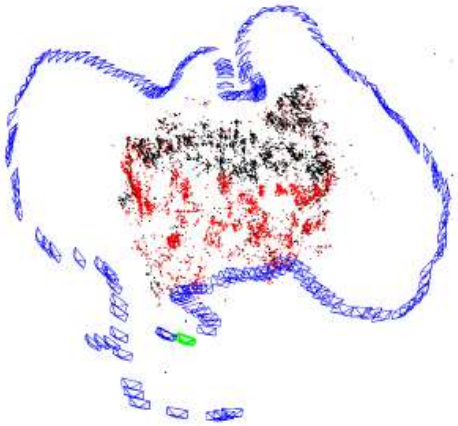


Fig. 1. ORB-SLAM system overview, showing all the steps performed by the tracking, local mapping, and loop closing threads. The main components of the place recognition module and the map are also shown.

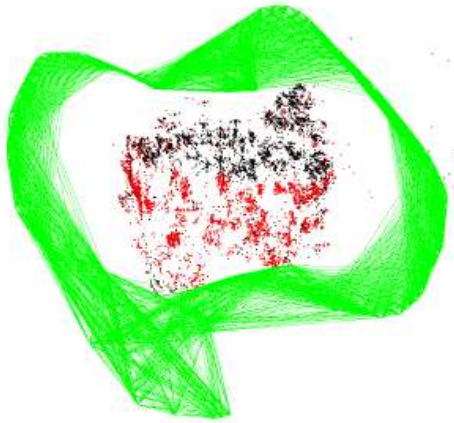
- На новом кадре находятся ORB точки
- Они сопоставляются с предыдущим кадром, и оценивается поза камеры.
- Если с предыдущим кадром не удается сопоставится, тогда релокализация
- Поскольку ORB ненадёжный, то точек сопоставляется мало, поэтому отдельный этап Track Local Map
- С предыдущих ключевых кадров видимые 3Д точки сцены проецируются на текущий кадр, фильтруются по эвристикам, и затем матчатся с несматченными ORB точками
- По итогу поза уточняет, и принимается решение, добавлять или нет новый ключевой кадр в карту

R.Mur-Artal, J.M.M.Montiel, J.D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on robotics, 2015

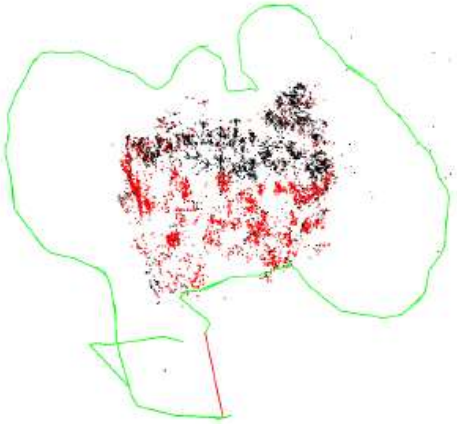
ORBSLAM - Mapping



(a) KeyFrames (blue), Current Camera (green), MapPoints (black, red), Current Local MapPoints (red)



(b) Covisibility Graph



(c) Spanning Tree (green) and Loop Closure (red)

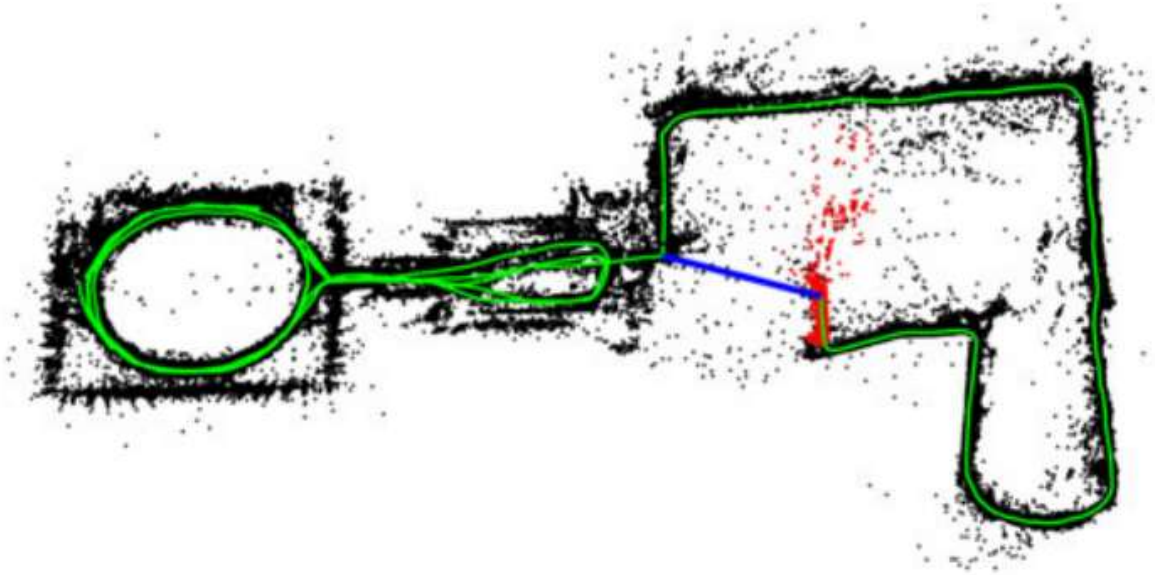


(d) Essential Graph

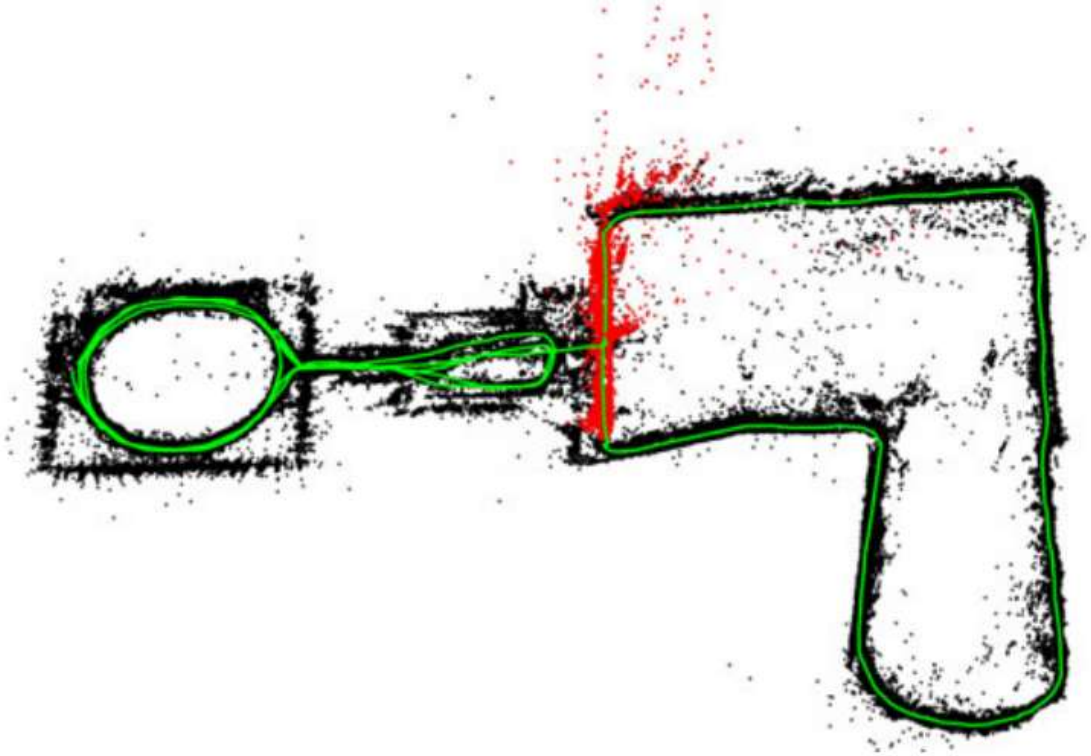
- Полная карта сцены – Co-visibility Graph
  - узлы это ключевые кадры
  - ребра связывают кадры, которые видят >= 15 одних и тех же точек
- Essential Graph – упрощённая карта сцены с прореженными связями, минимальное оставное дерево
- EG используется, когда нужно уточнить всю карту при закрытии циклов
- Ключевые кадры выбираются часто, предусмотрен механизм отбраковки лишних ключевых кадров
- Поиск похожих изображений по тем же ключевым точкам используется для детекции циклов и ре-локализации
  - https://github.com/dorian3d/DBoW2

ORBSLAM - Loop Closure

Закрытие цикла приводит к «скачкам» камеры и пересчёту карты

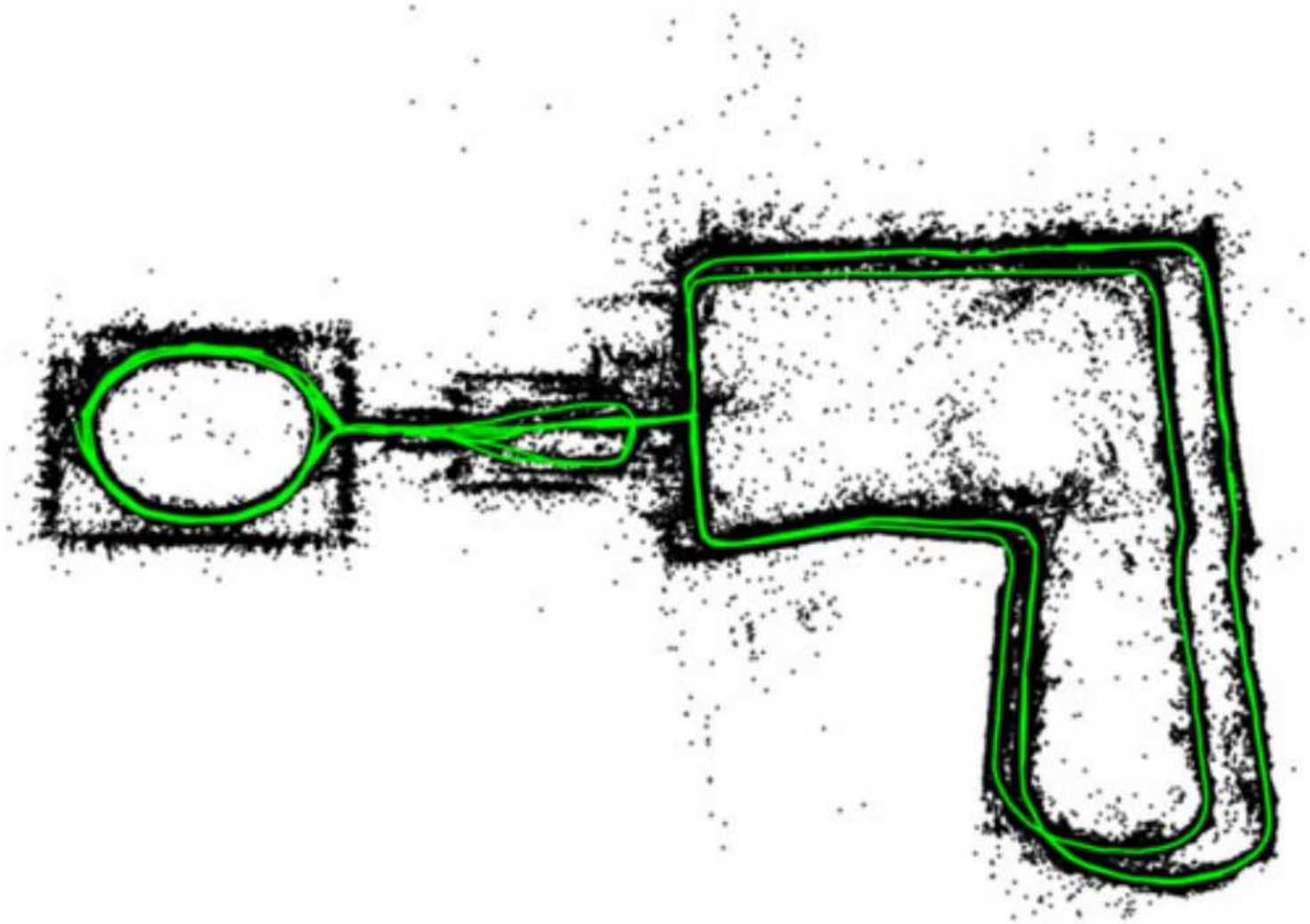


После закрытия цикла



До закрытия цикла

ORBSLAM - Ограничения закрытия циклов



Движение было в одном и другом направлении, поэтому картинки не были похожими, цикл не был обнаружен, и закрытие цикла не произошло

Скорости работы

TABLE I
TRACKING AND MAPPING TIMES IN NEWCOLLEGE

Thread	Operation	Median (ms)	Mean (ms)	Std (ms)
TRACKING	ORB extraction	11.10	11.42	1.61
	Initial Pose Est.	3.38	3.45	0.99
	Track Local Map	14.84	16.01	9.98
	Total	30.57	31.60	10.39
LOCAL MAPPING	KeyFrame Insertion	10.29	11.88	5.03
	Map Point Culling	0.10	3.18	6.70
	Map Point Creation	66.79	72.96	31.48
	Local BA	296.08	360.41	171.11
	KeyFrame Culling	8.07	15.79	18.98
	Total	383.59	464.27	217.89

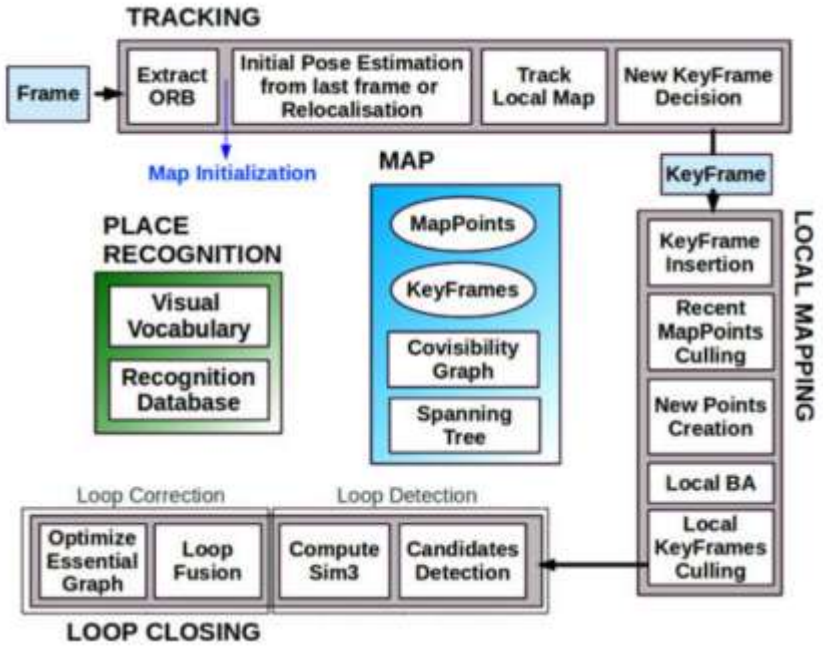
Скорости работы

TABLE II
LOOP CLOSING TIMES IN NEWCOLLEGE

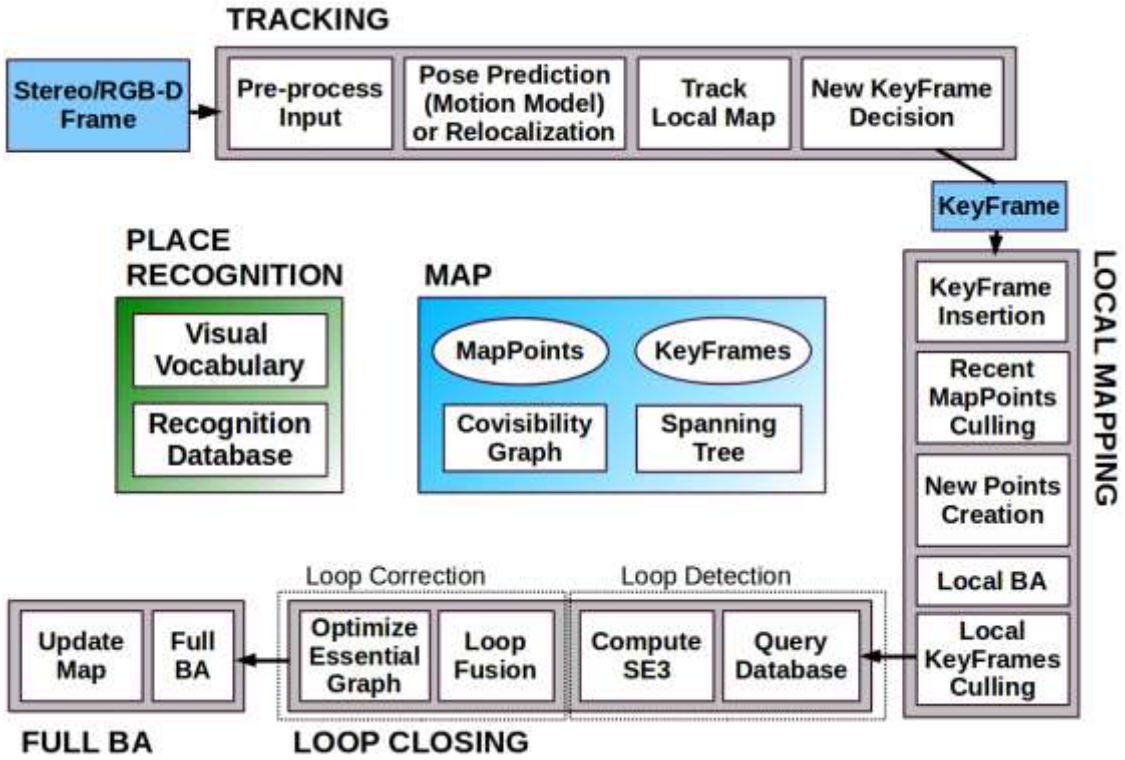
			Loop Detection (ms)		Loop Correction (s)		
Loop	KeyFrames	Essential Graph Edges	Candidates Detection	Similarity Transformation	Fusion	Essential Graph Optimization	Total (s)
1	287	1347	4.71	20.77	0.20	0.26	0.51
2	1082	5950	4.14	17.98	0.39	1.06	1.52
3	1279	7128	9.82	31.29	0.95	1.26	2.27
4	2648	12547	12.37	30.36	0.97	2.30	3.33
5	3150	16033	14.71	41.28	1.73	2.80	4.60
6	4496	21797	13.52	48.68	0.97	3.62	4.69

ORBSLAM2

Развитие ORBSLAM с добавлением уточнения всей карты (глобального BA) и работой со стерео-данными и RGB-D



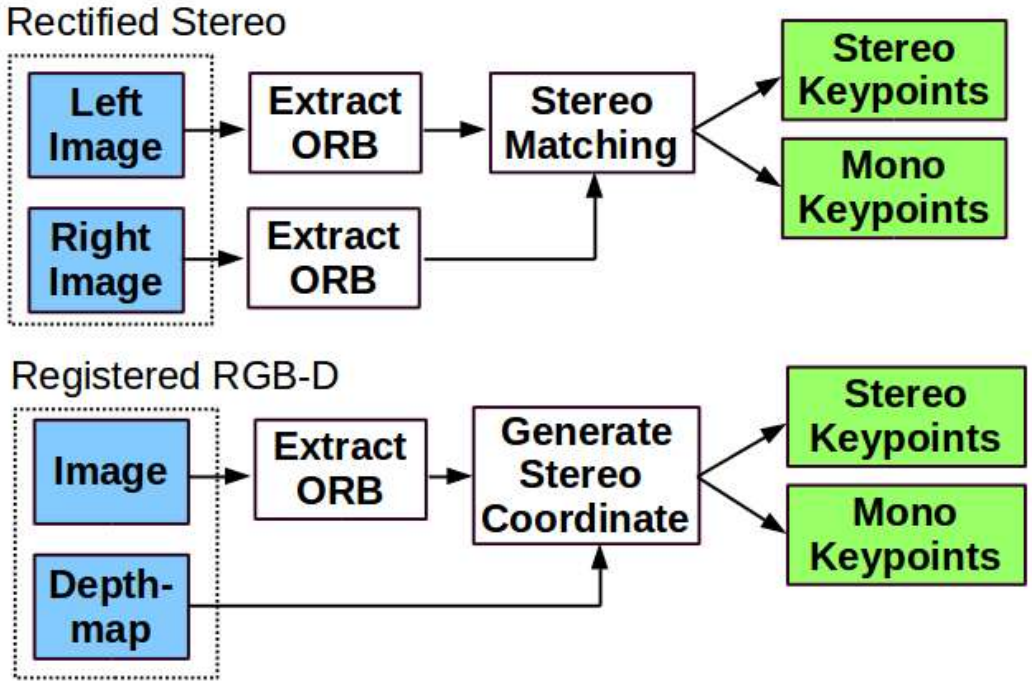
ORBSLAM



ORBSLAM2

Raúl Mur-Artal and Juan D. Tardós. **ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras**. *IEEE Transactions on Robotics,* vol. 33, no. 5, pp. 1255-1262, 2017.

ORBSLAM2



- На стереопаре мы сопоставляем точки между ракурсами, триангулируем, и сразу получаем 3Д координаты
- Выделяем «близкие» и «дальние» точки.
- Для близких достаточный параллакс позволяет точно оценить глубину и использовать для оценки позы камеры
- Дальние точки оцениваем только примерно, можем использовать для оценки поворотов

Raúl Mur-Artal and Juan D. Tardós. **ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras**. *IEEE Transactions on Robotics,* vol. 33, no. 5, pp. 1255-1262, 2017.

Влияние стерео

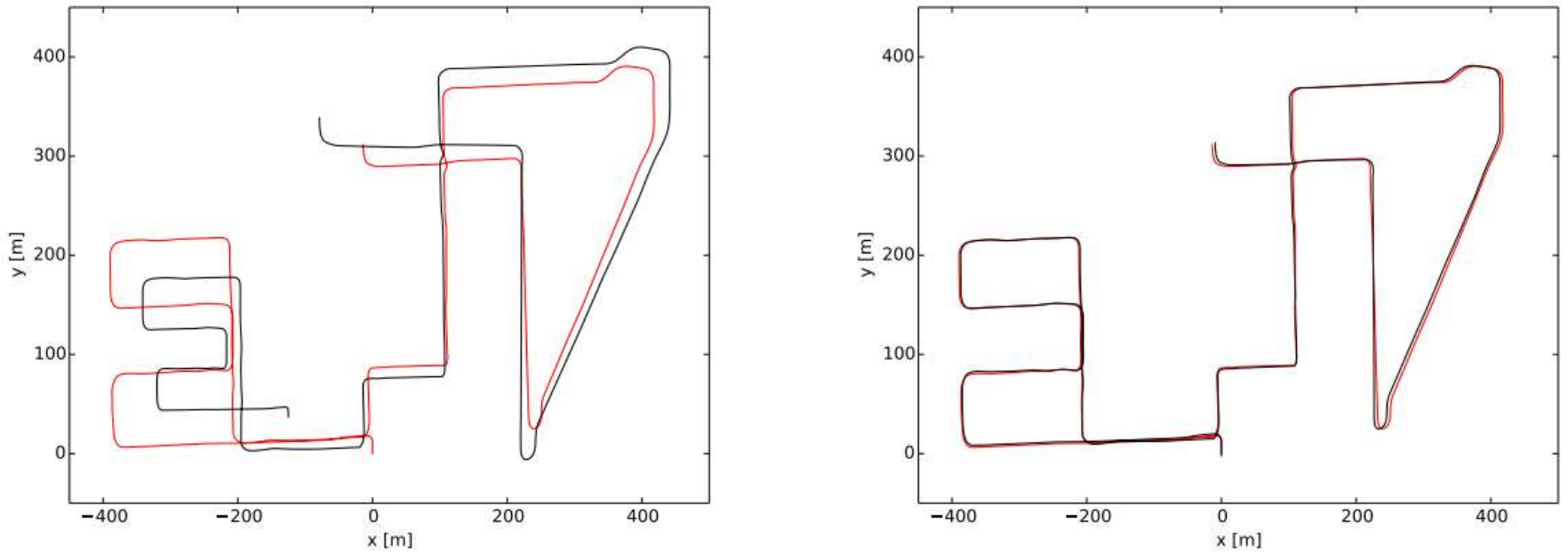
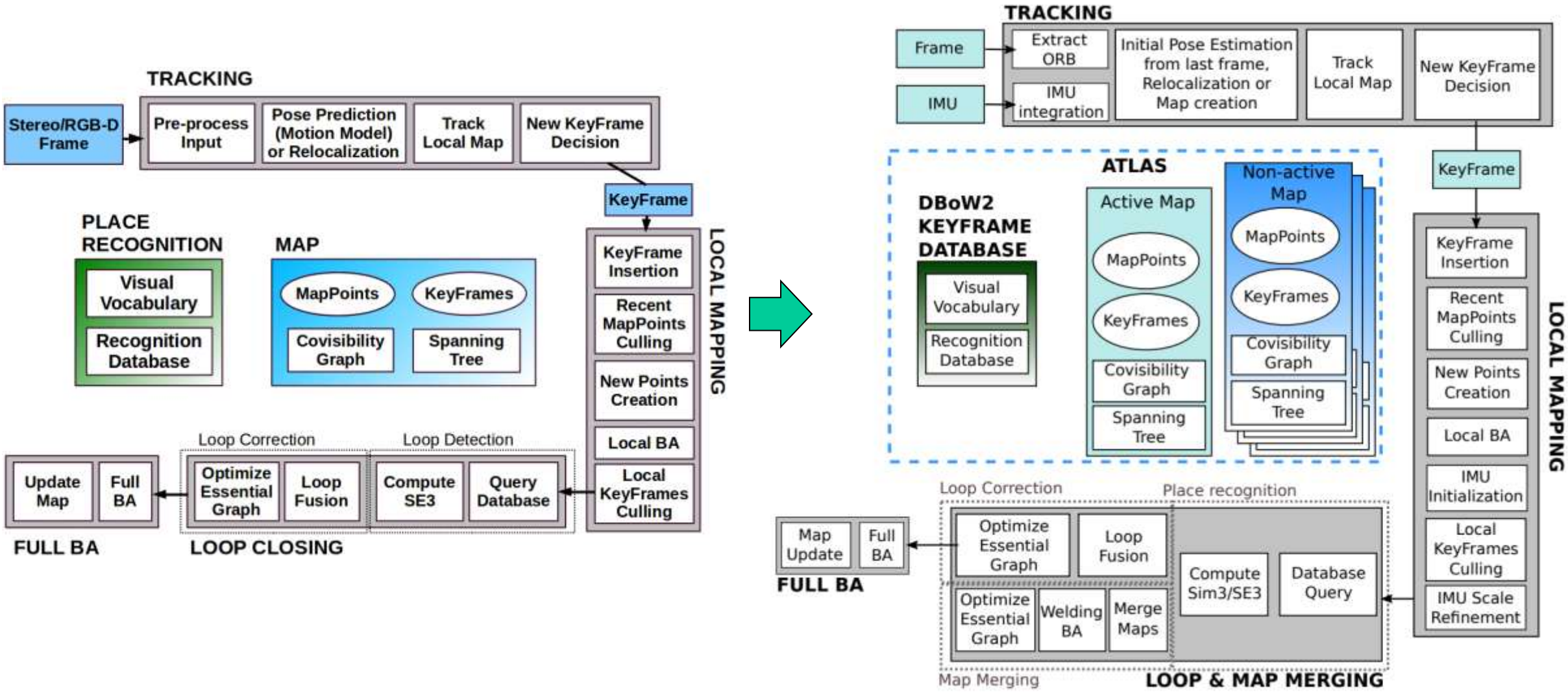


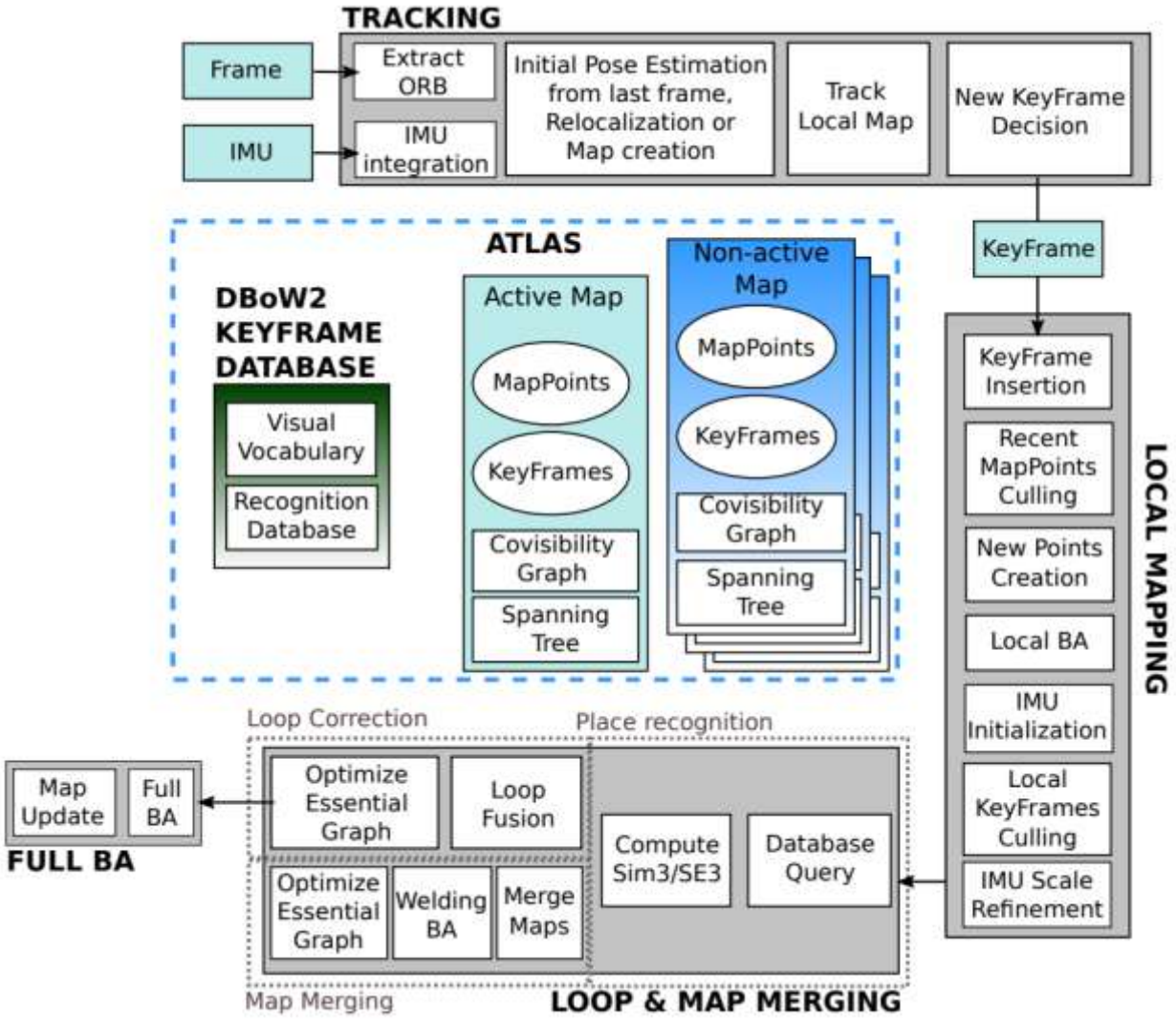
Fig. 5. Estimated trajectory (black) and ground-truth (red) in KITTI 08. Left: monocular ORB-SLAM [1], right: ORB-SLAM2 (stereo). Monocular ORB-SLAM suffers from severe scale drift in this sequence, especially at the turns. In contrast the proposed stereo version is able to estimate the true scale of the trajectory and map without scale drift.

ORBSLAM3



https://github.com/UZ-SLAMLab/ORB_SLAM3

ORBSLAM3



- Акселерометры (IMU) есть во многих устройствах (телефонах, камерах)
- IMU позволяет улучшить трекинг камеры при motion blur, малом числе ORB, и в целом повысить точность
- Atlas – карта, состоящая из многих отдельных карт. Одна из них «активна», используется для текущего трекинга.
- Если обнаруживаем «связь» между картами в атласе, тогда мы их объединяем

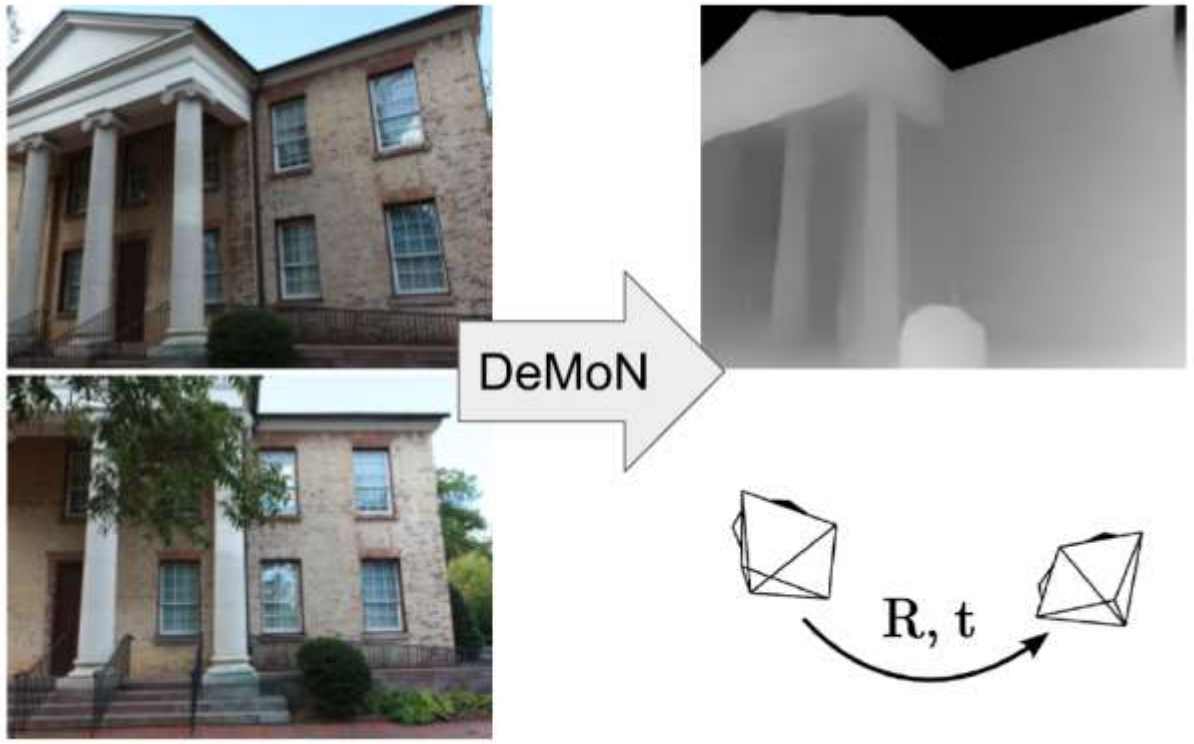
ORBSLAM3

Table II: Performance comparison in the EuRoC dataset (RMS ATE in m., scale error in %). Except where noted, we show results reported by the authors of each system, for all the frames in the trajectory, comparing with the processed GT.

			MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg[1]
Monocular	ORB-SLAM [4]	ATE[2,3]	0.071	0.067	0.071	0.082	**0.060**	**0.015**	0.020	-	**0.021**	**0.018**	-	0.047*
	DSO [27]	ATE	0.046	0.046	0.172	3.810	0.110	0.089	0.107	0.903	0.044	0.132	1.152	0.601
	SVO [24]	ATE	0.100	0.120	0.410	0.430	0.300	0.070	0.210	-	0.110	0.110	1.080	0.294*
	DSM [31]	ATE	0.039	0.036	0.055	**0.057**	0.067	0.095	0.059	0.076	0.056	0.057	**0.784**	**0.126**
	ORB-SLAM3 (ours)	ATE	**0.016**	**0.027**	**0.028**	0.138	0.072	0.033	**0.015**	**0.033**	0.023	0.029	-	0.041*
Stereo	ORB-SLAM2 [3]	ATE	0.035	**0.018**	0.028	0.119	0.060	**0.035**	**0.020**	**0.048**	**0.037**	0.035	-	0.044*
	VINS-Fusion [44]	ATE	0.540	0.460	0.330	0.780	0.500	0.550	0.230	-	0.230	0.200	-	0.424*
	SVO [24]	ATE	0.040	0.070	0.270	0.170	0.120	0.040	0.040	0.070	0.050	0.090	0.790	0.159
	ORB-SLAM3 (ours)	ATE	**0.029**	0.019	**0.024**	**0.085**	**0.052**	**0.035**	0.025	0.061	0.041	**0.028**	**0.521**	**0.084**
Monocular Inertial	MCSKF [33]	ATE[5]	0.420	0.450	0.230	0.370	0.480	0.340	0.200	0.670	0.100	0.160	1.130	0.414
	OKVIS [39]	ATE[5]	0.160	0.220	0.240	0.340	0.470	0.090	0.200	0.240	0.130	0.160	0.290	0.231
	ROVIO [42]	ATE[5]	0.210	0.250	0.250	0.490	0.520	0.100	0.100	0.140	0.120	0.140	0.140	0.224
	ORBSLAM-VI [4]	ATE[2,3] scale error[2,3]	0.075 0.5	0.084 0.8	0.087 1.5	0.217 3.5	0.082 0.5	**0.027** 0.9	0.028 0.8	- -	**0.032** 0.2	0.041 1.4	0.074 0.7	0.075* 1.1*
	VINS-Mono [7]	ATE[4]	0.084	0.105	0.074	0.122	0.147	0.047	0.066	0.180	0.056	0.090	0.244	0.110
	VI-DSO [46]	ATE scale error	**0.062** 1.1	0.044 0.5	0.117 0.4	0.132 0.2	0.121 0.8	0.059 1.1	0.067 1.1	0.096 0.8	0.040 1.2	0.062 0.3	0.174 0.4	0.089 0.7
	ORB-SLAM3 (ours)	ATE scale error	**0.062** 1.4	**0.037** 0.3	**0.046** 0.8	**0.075** 0.5	**0.057** 0.3	0.049 2.0	**0.015** 0.6	**0.037** 2.2	0.042 0.7	**0.021** 0.4	**0.027** 1.0	**0.043** 0.9
Stereo Inertial	VINS-Fusion [44]	ATE[4]	0.166	0.152	0.125	0.280	0.284	0.076	0.069	0.114	0.066	0.091	0.096	0.138
	BASALT [47]	ATE[3]	0.080	0.060	0.050	0.100	**0.080**	0.040	0.020	0.030	**0.030**	0.020	-	0.051*
	Kimera [8]	ATE	0.080	0.090	0.110	0.150	0.240	0.050	0.110	0.120	0.070	0.100	0.190	0.119
	ORB-SLAM3 (ours)	ATE scale error	**0.036** 0.6	**0.033** 0.2	**0.035** 0.6	**0.051** 0.2	0.082 0.9	**0.038** 0.8	**0.014** 0.6	**0.024** 0.8	0.032 1.1	**0.014** 0.2	**0.024** 0.2	**0.035** 0.6

Нейросетевой SLAM

Deep Visual Odometry

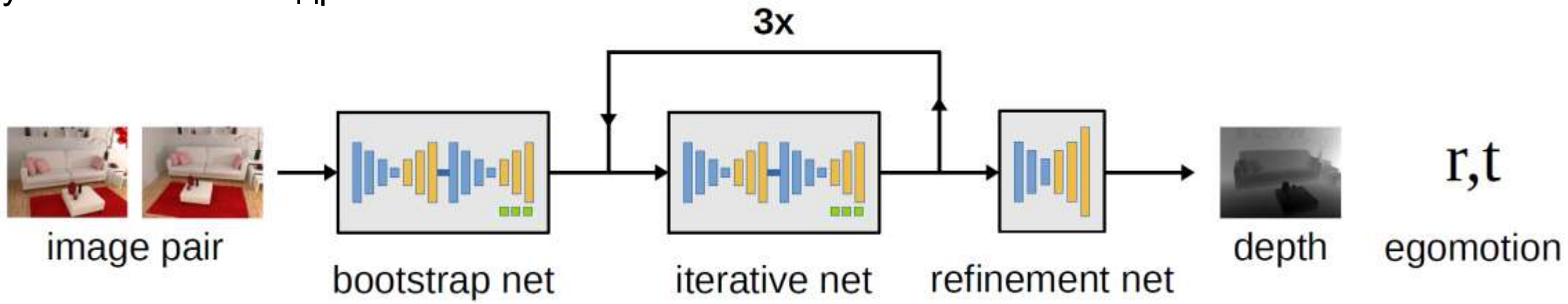


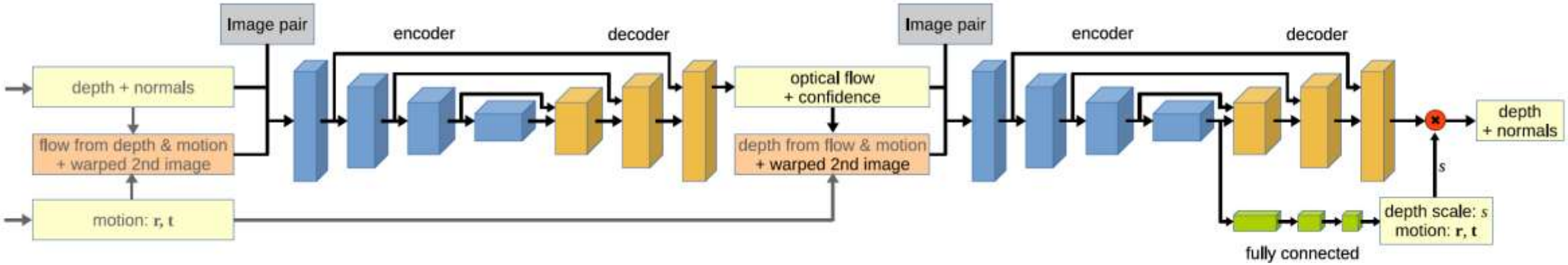
- Нейросетевые методы начали проникать в задачу SLAM постепенно, рассматривая отдельные подзадачи
- Пример - Visual Odometry
- Задача – оценить движение камеры между парой кадров
- Вспомогательные задачи, такие как оценка оптического потока, оценка карты глубины и т.д. используются как вспомогательные, чтобы нейросеть получала достаточно информации для обучения

Ummenhofer et. al. DeMoN: Depth and Motion Network for Learning Monocular Stereo. ICCV2017

DeMoN: Depth & Motion Network

Идея: чередуем оценку оптического потока и карты глубины для того, чтобы заставить сеть учитывать оба кадра



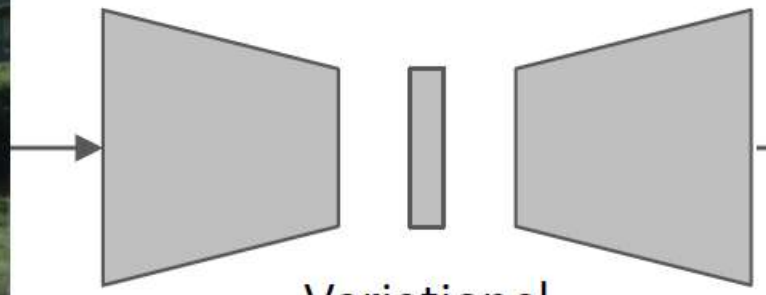


Ummenhofer et. al. DeMoN: Depth and Motion Network for Learning Monocular Stereo. ICCV2017

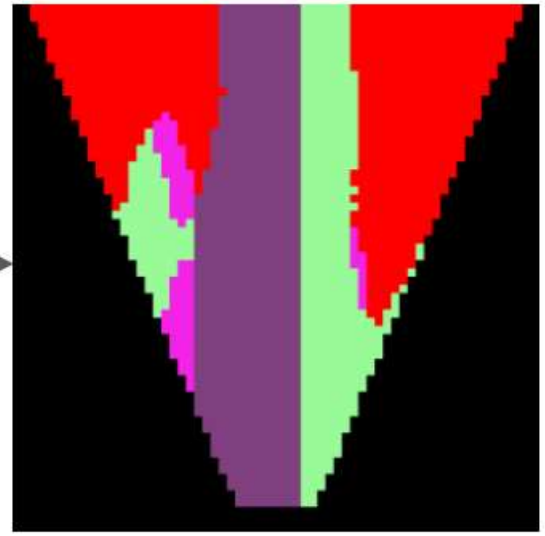
Deep mapping

Генерация семантической карты напрямую из 2D изображений





Variational
Encoder-Decoder Network



Input:
Monocular
RGB Image

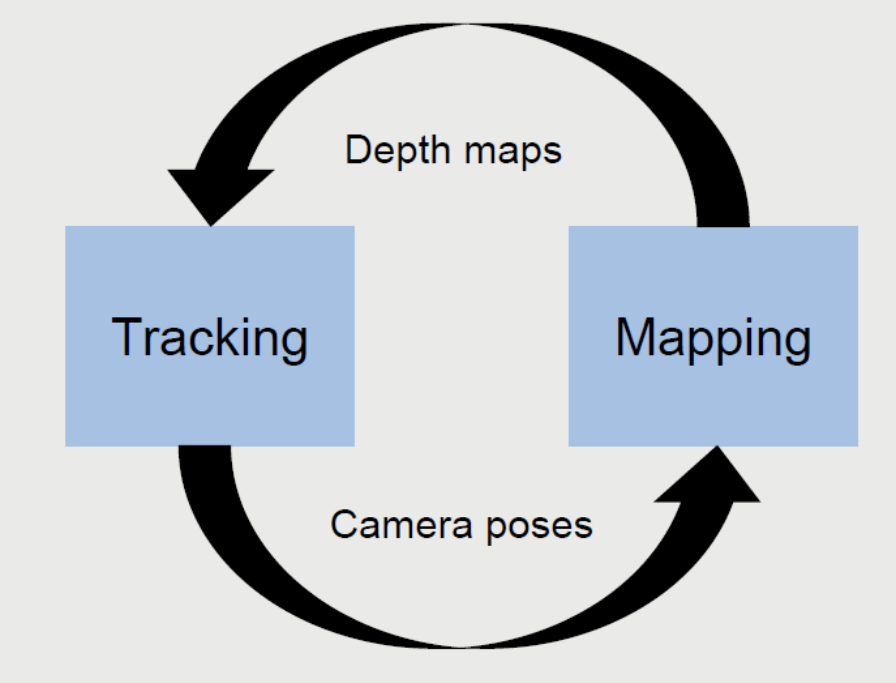
Legend:	Out of FOV
Road	Sidewalk
Terrain	Non free-space

Output:
Semantic-metric
Occupancy Grid Map

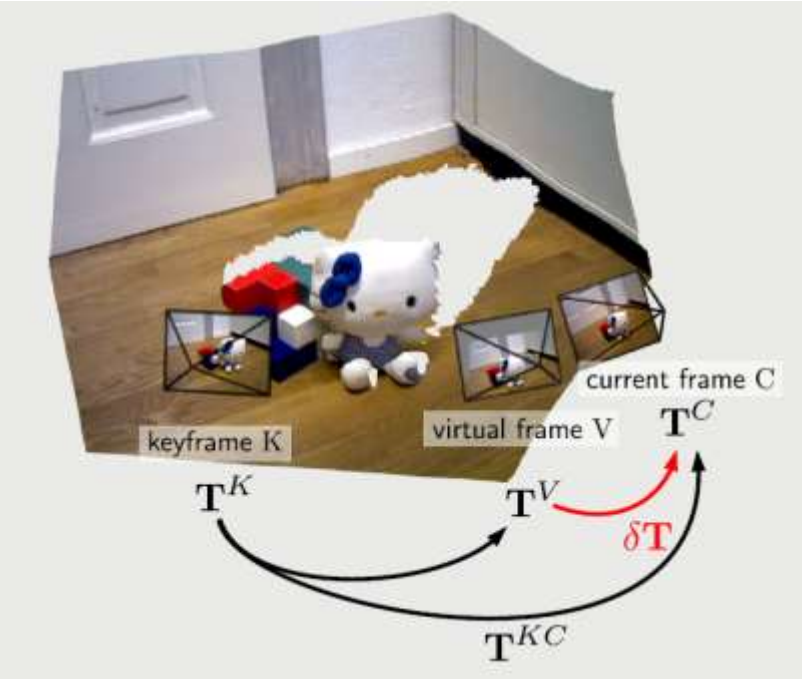
Source: arXiv:1804.02176v2

Deep Tracking and Mapping (DeepTAM)

Оценка позы на основе ключевых кадров и построение карты. Без loop closure и relocalization



Схема

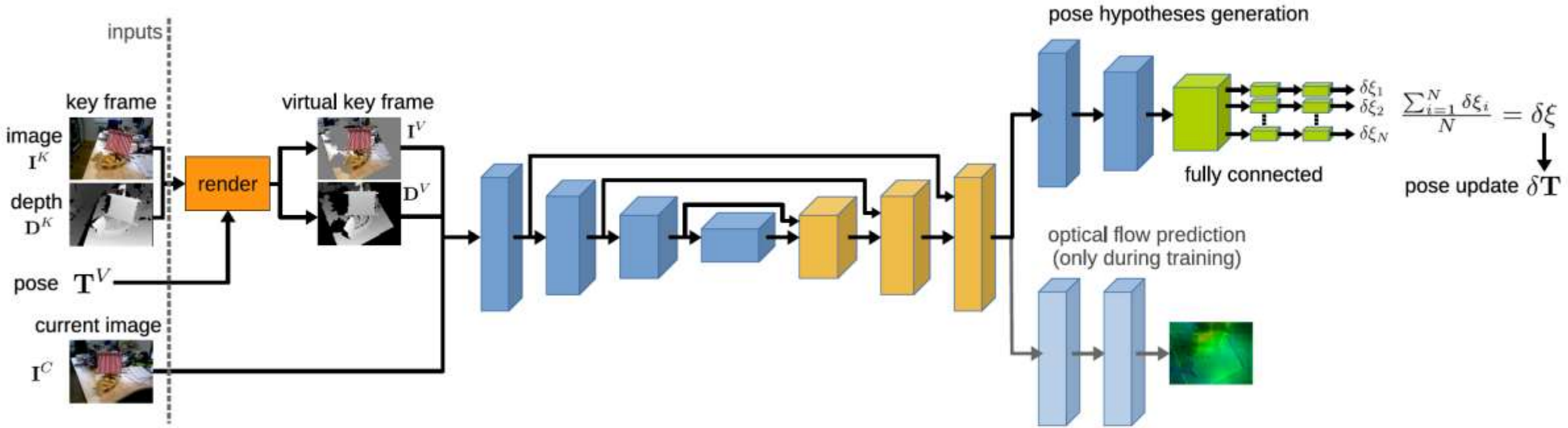


Оцениваем позу камеры относительно предсказанного виртуального кадра

H. Zhou et al., DeepTAM: Deep Tracking and Mapping. ECCV 2018

Модуль трекинга

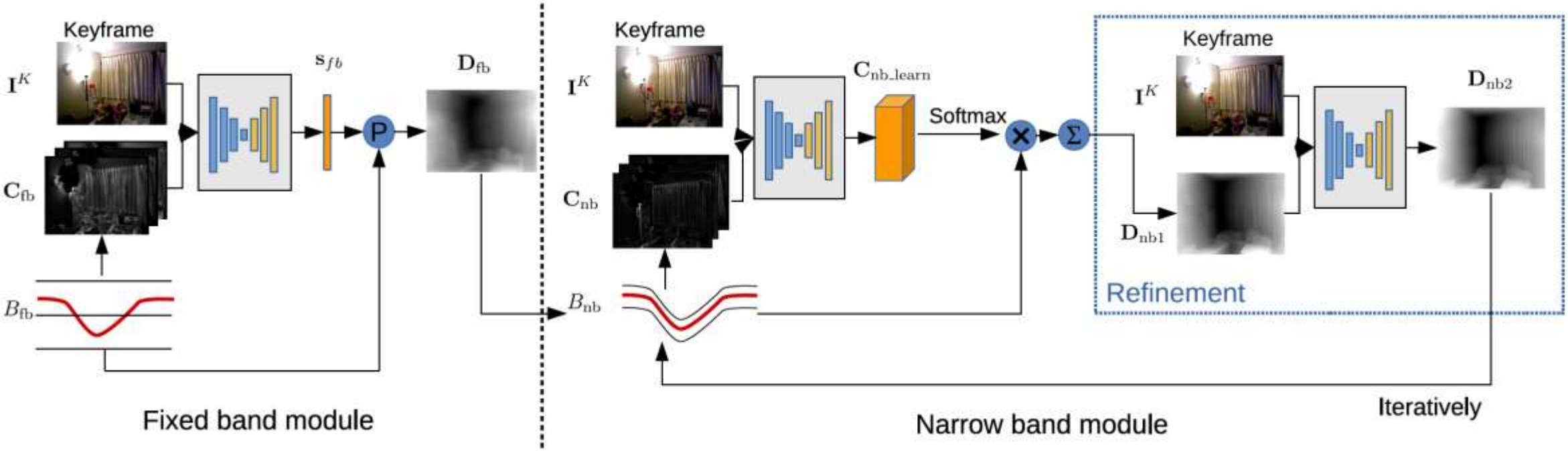
- Оцениваем позу и оптический поток
- Генерируем набор гипотез и затем усредняем их



H. Zhou et al., DeepTAM: Deep Tracking and Mapping. ECCV 2018

Модуль картографирования

- Mapping = high-quality depth estimation for keyframe



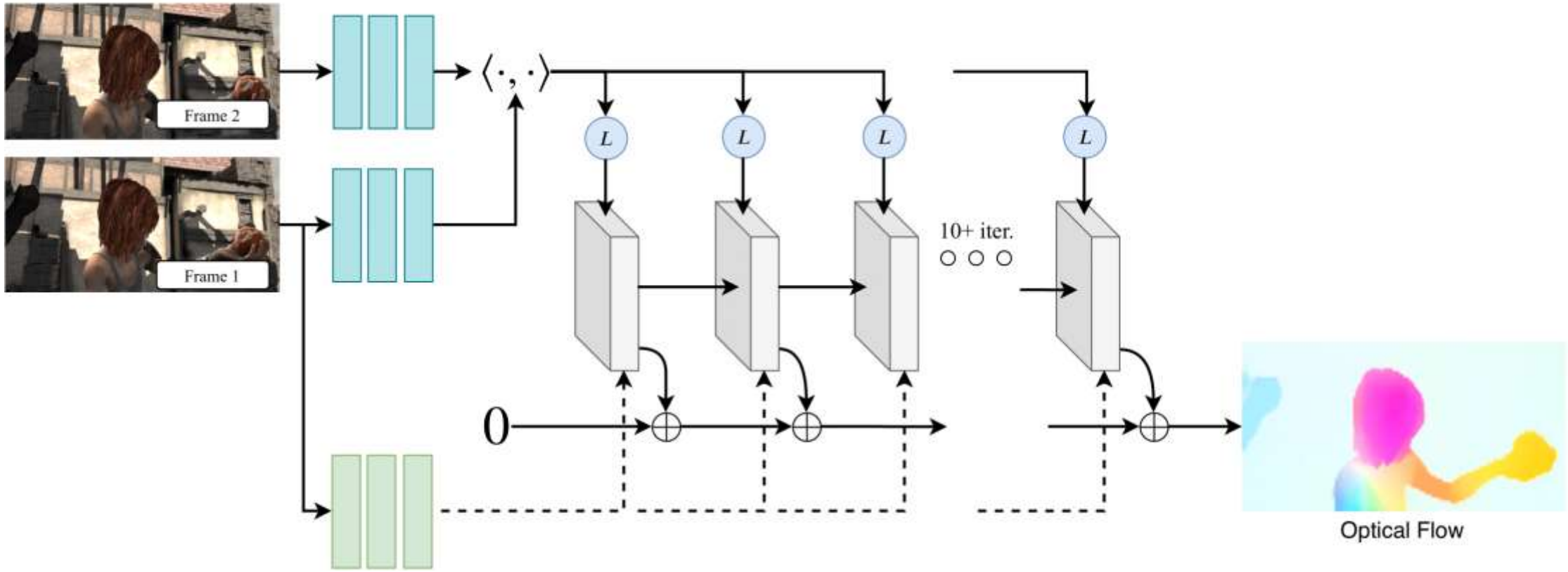
H. Zhou et al., DeepTAM: Deep Tracking and Mapping. ECCV 2018

Пример работы

H. Zhou et al., DeepTAM: Deep Tracking and Mapping. ECCV 2018

Recurrent All-Pairs Field Transforms (RAFT)

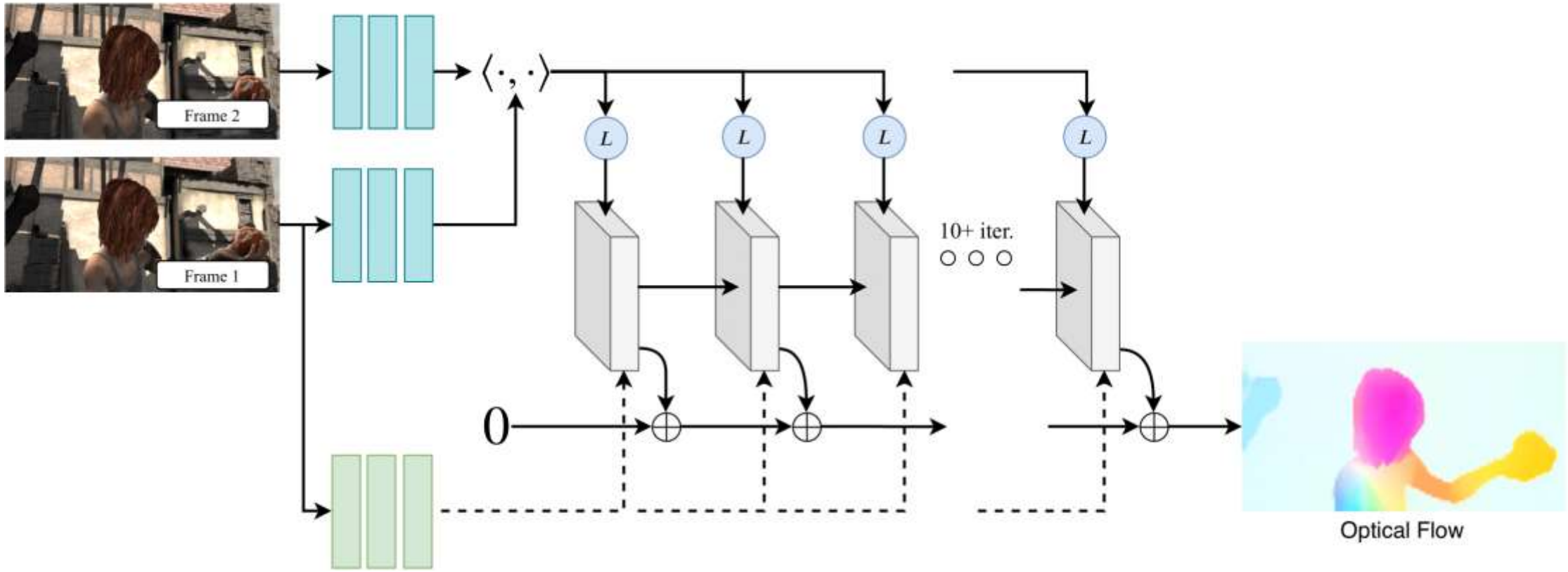
Метод оценки оптического потока на основе рекуррентного модуля



Zachary Teed, Jia Deng RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. ECCV 2020

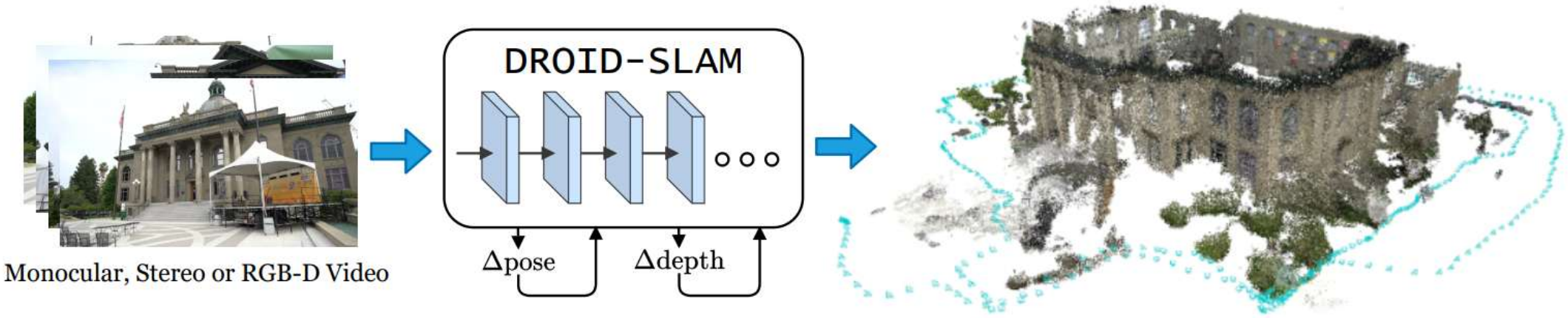
Recurrent All-Pairs Field Transforms (RAFT)

Метод оценки оптического потока на основе рекуррентного модуля



Zachary Teed, Jia Deng RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. ECCV 2020

DROID-SLAM



- Полная схема SLAM, вдохновлённая идеями RAFT
- Будем строить граф сцены из набора ключевых кадров, для каждого оценивая позу и карту глубины.
- Нейросетевой модуль применяется к рёбрам графа, уточняя позу и карту глубины кадра

https://github.com/princeton-vl/DROID-SLAM

DROID-SLAM



- Система состоит из двух потоков – Frontend и Backend
- Frontend работает с потоком кадров.
  - Для нового кадра считают признаки, связывают с 3мя ближайшими ключевыми кадрами по оптическому потоку
  - Поза инициализируется линейной моделью
  - Оператор уточнения применяется несколько раз для уточнения позы и глубины нового ключевого кадра
- Backend – глобальный BA на всём графе сцены
  - Перестраиваем граф сцены, посчитав оптический поток между всеми парами кадров, инициализировав матрицу расстояний N*N
  - Проходим по всем рёбрам, начиная от близких по времени, и затем выбирая новые кадры в порядке увеличения оптического потока, подавляя близкие рёбра
  - Применяем оператор уточнения для всех ребёр получившегося графа

https://github.com/princeton-vl/DROID-SLAM

Резюме SLAM

- Задача SLAM близка к задаче SFM, но работает в итеративном он-лайн режиме, в реальном времени
- На практике пока в основном используются мультимодальные методы, работающие по классическому подходу с ключевыми точками
- Нейросетевые методы активно развиваются, превосходят по точности классические методы для случая RGB/RGBD камер, но пока очень вычислительно затратны