SLAM/VIO Tutorial

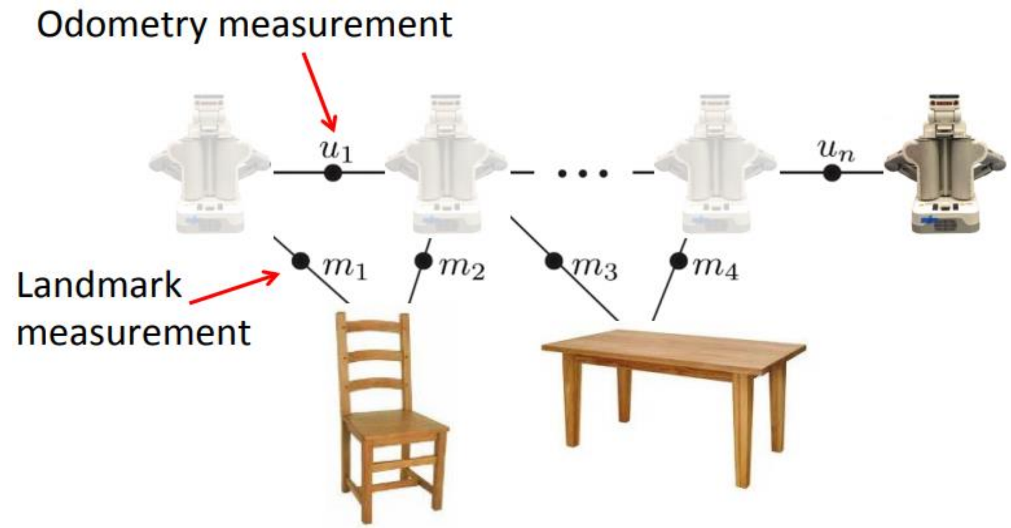
(Mostly on Front End)

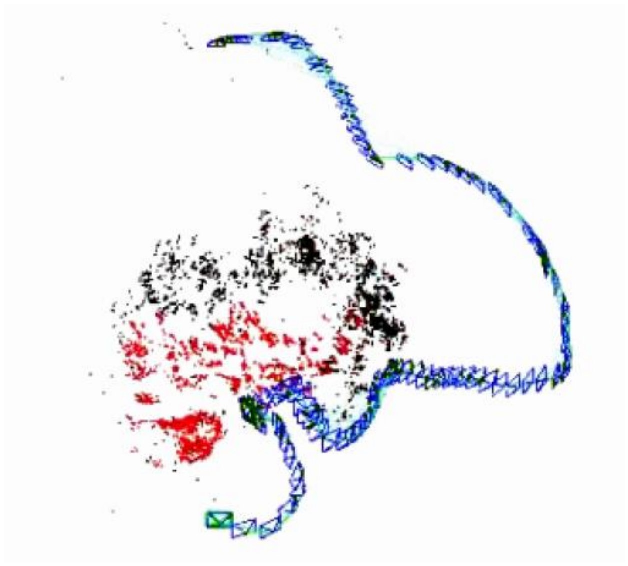
Zhou Yu
2020.06.18

National University of Singapore

- What is SLAM/VIO exactly?
- What’s the difference?
- How to formulate the problem?

What is SLAM?





Mapping: What is the world around me ?

Integration of the information gathered with sensors into a given representation.

- sense from various positions
- integrate measurements to produce map
- assumes perfect knowledge of position

Localization: Where am I in the world?

Estimation of the robot pose relative to a map

- sense
- relate sensor readings to a world model
- compute location relative to model
- assumes a perfect world model

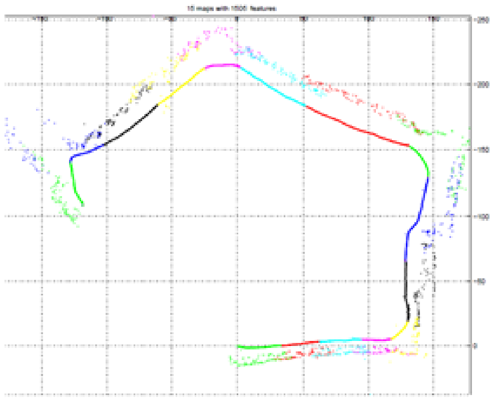
What is odometry?

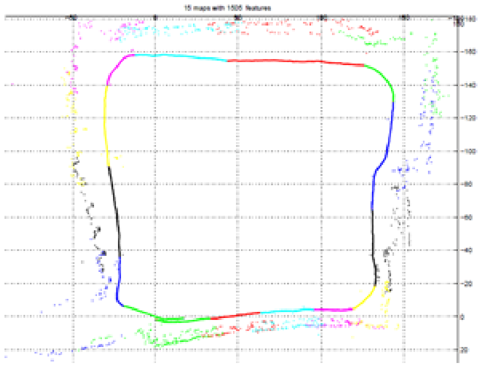
The process of incrementally estimating the pose of the vehicle by examining the changes that motion induces on sensor measurement, such as wheel, laser, IMU and Image.

Difference

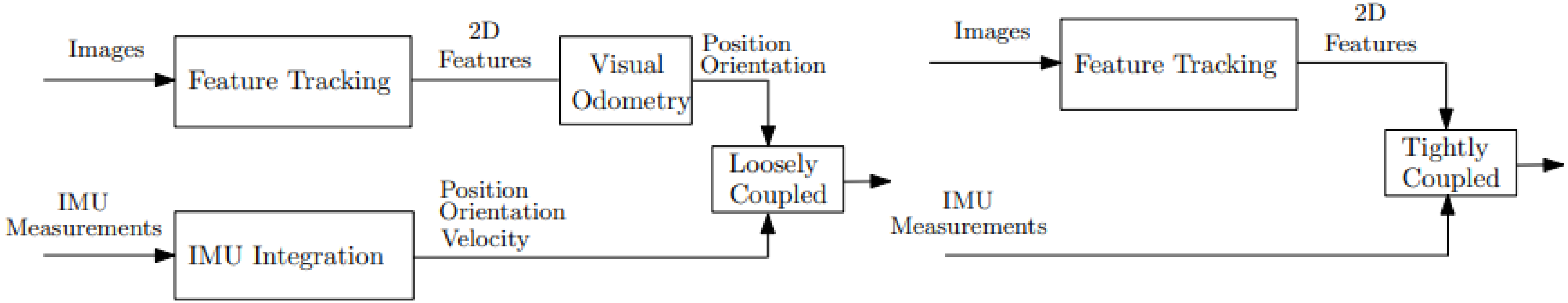
Odometry only aims to the local consistency of the trajectory, can be used as a building block of SLAM. It is SLAM before loop closures

Odometry trades off consistency for real-time performance, without the need to keep track of all the previous history of the camera.





Two paradigms of VIO



Comparison of loosely (left) and tightly coupled (right) paradigms for VIO

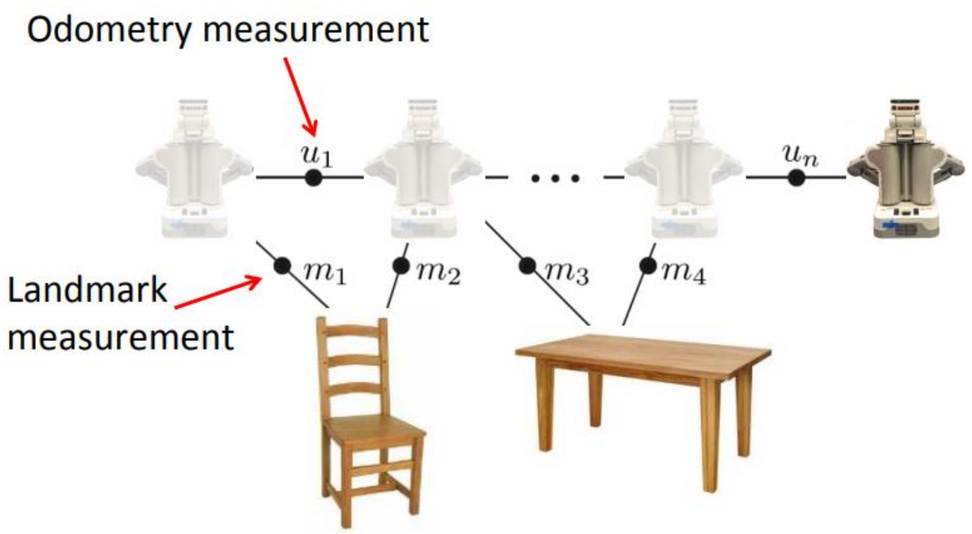
Loosely coupled methods:

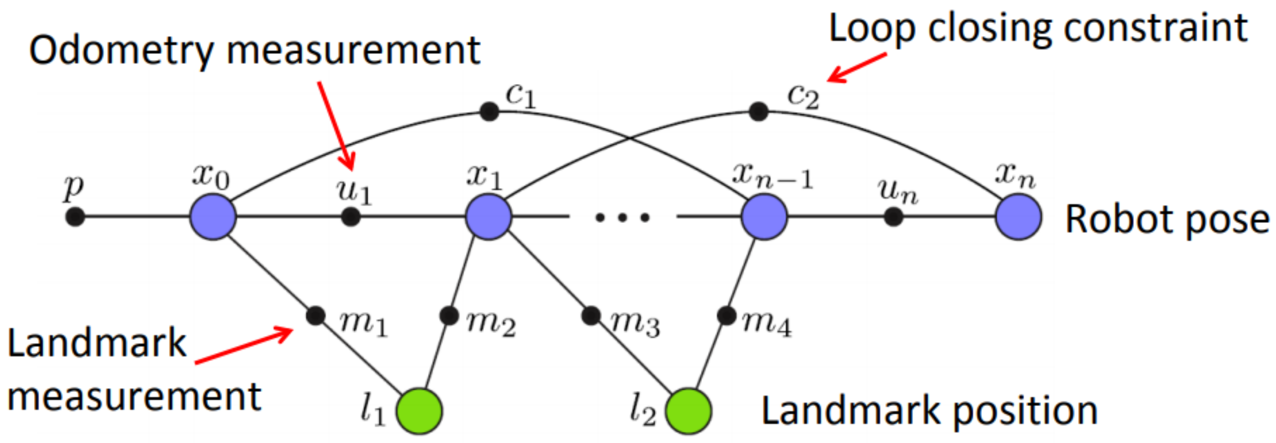
Process visual and inertial measurements separately and then fuse together. Incapable of correcting drift in the vision-only estimator

Tightly coupled methods:

Compute the final output directly from the raw camera and IMU measurements. More accurate

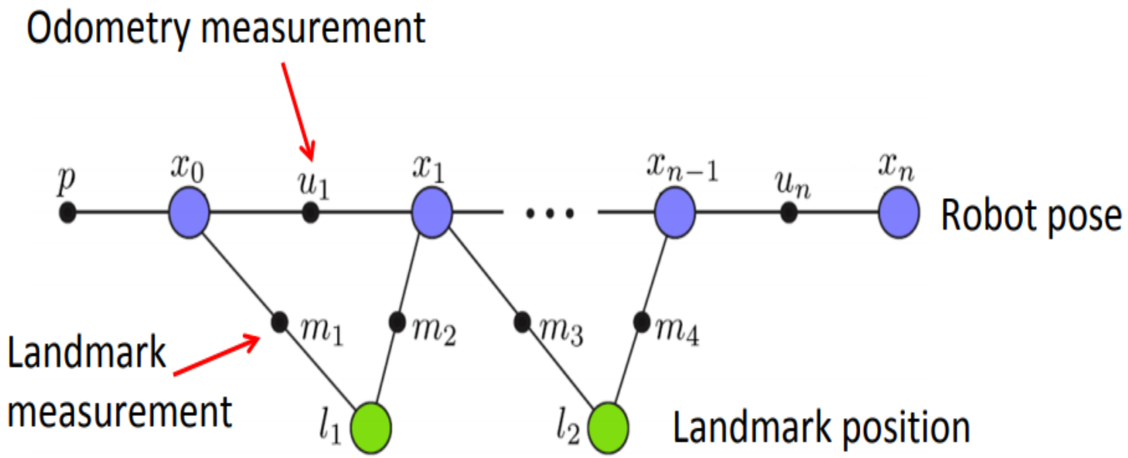
Problem formation --- SLAM/VIO





Bipartite graph with variable nodes and factor nodes

Problem formation --- SLAM/VIO



$$\mathbf{X}^* := \operatorname{argmax}_{\mathbf{X}} P(\mathbf{Y}|\mathbf{X})$$

Maximum Likelihood: find the model parameters that maximize the probability of obtaining the actual measurements.

X: State

- 6 DOF position & orientation (**pose**)
- 3 DOF landmarks or depth in a reference frame (**map**)

Y: Observation

- Geometry measurement (**Indirect**) or Photometric measurement (**Direct**)
- IMU preintegration

If assume Gaussian noise, then SLAM/VIO can be seen as a **Sparse Least-Squares optimization Problem.**

$$E = \sum_{\substack{i \in \mathcal{P} \\ t \in \mathrm{obs}(i)}} \mathbf{r}_{it}^{\top} \mathbf{\Sigma}_{it}^{-1} \mathbf{r}_{it} + \sum_{(i,j) \in \mathcal{C}} \mathbf{r}_{ij}^{\top} \mathbf{\Sigma}_{ij}^{-1} \mathbf{r}_{ij}$$

- What are the states, map and observations specifically?
- What are the IMU preintegration, geometry and photometric error?

State --- position & orientation

VIO is the process of estimating the **state** of the sensor suite using the camera and IMU measurements. Typically, the quantities to estimate are **N states at different times**.

$$\mathbf{X}_i = [\mathbf{T}_{WI}^i, \mathbf{v}_{WI}^i, \mathbf{b}_a^i, \mathbf{b}_g^i], \quad i = 1, 2, 3, \ldots, N$$

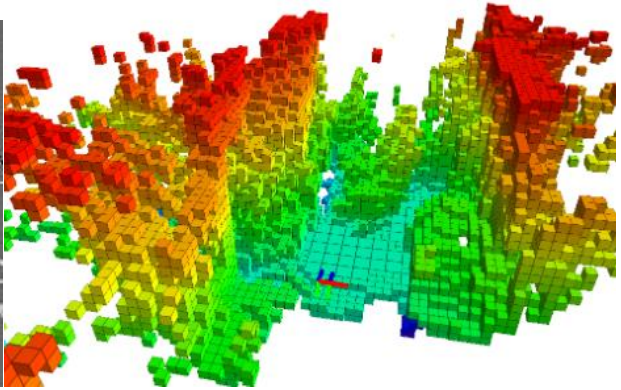
where **T** is the 6-DoF pose of the vehicle, **v** is the velocity of the vehicle, ba and bg are the biases of the accelerometer and gyroscope respectively.

-**biases** are necessary for computing **the actual sensor angular velocity and acceleration** from the raw measurements

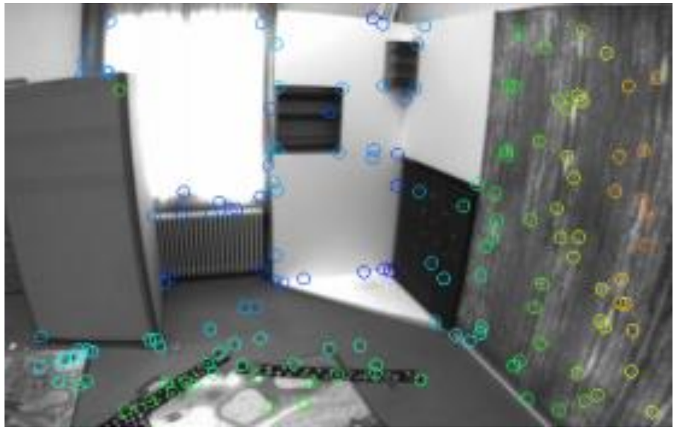
-**velocity** is needed for **integrating acceleration** to get position.

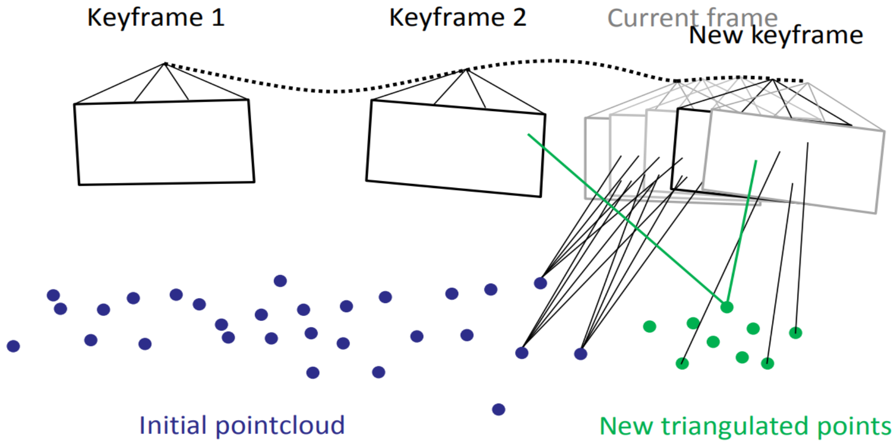
map

What is the map in VIO?



Interesting points in environment





Observation --- IMU preintegration

What is IMU Preintegration

Reparametrization of the relative motion constraints from IMU measurements integrated **between frames**. Repeated integration when the state estimate changes can be avoided by the Preintegration.

Why do we need IMU preintegration?

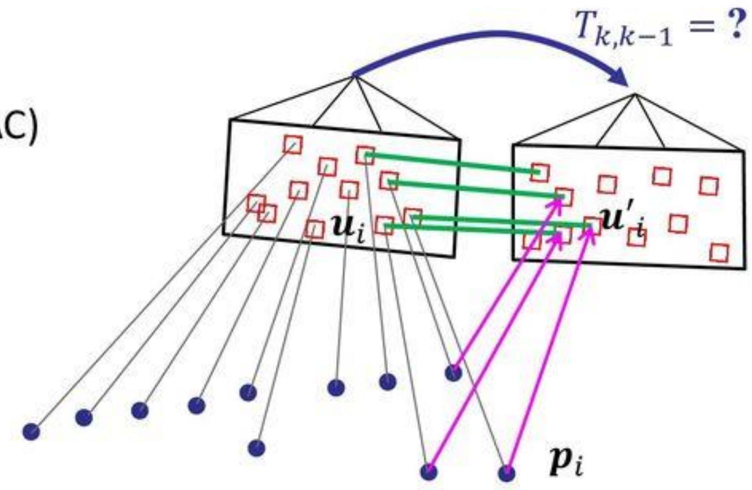
It is infeasible for real-time applications to add a state at every IMU measurement, the problem complexity grows with the dimension of the states. So we group the IMU measurements between image frames to form a **pseudo super measurement**.

Forster, Christian, et al. "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." Georgia Institute of Technology, 2015.

Observation
Geometry/Photometric measurement

Feature-based methods

1. Extract & match features (+RANSAC)
2. Minimize **Reprojection error** minimization



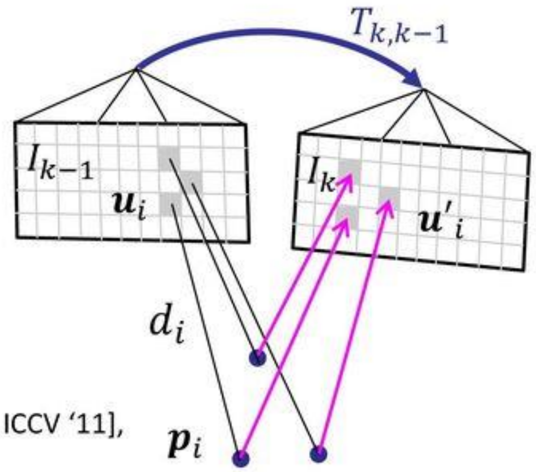
$$T_{k,k-1} = \arg\min_{T} \sum_i \| \boldsymbol{u'}_i - \pi(\boldsymbol{p}_i) \|^2_{\Sigma}$$

Direct methods

1. Minimize **photometric error**

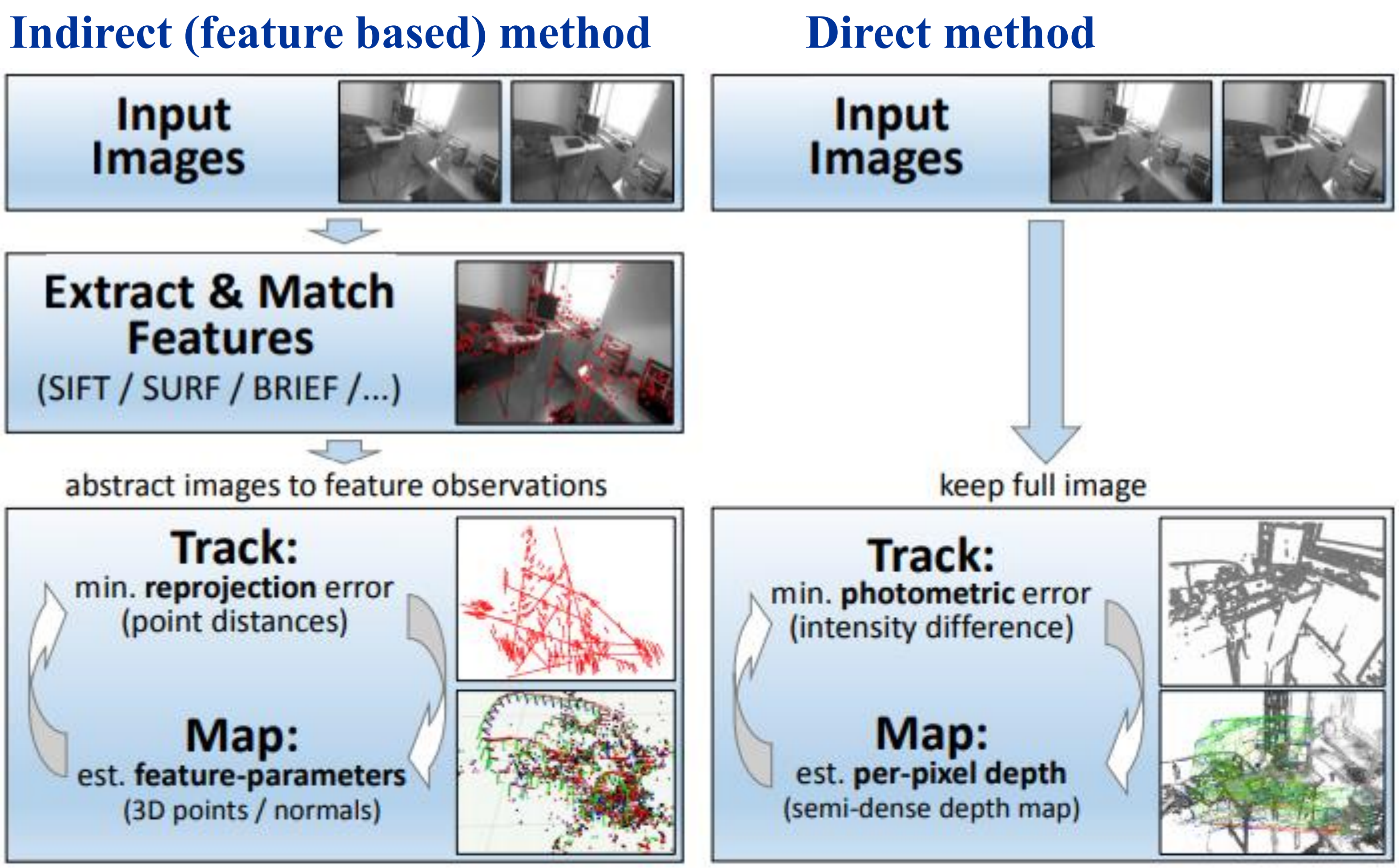
$$T_{k,k-1} = \arg\min_{T} \sum_i \| I_k(\boldsymbol{u'}_i) - I_{k-1}(\boldsymbol{u}_i) \|^2_{\sigma}$$

where $\boldsymbol{u'}_i = \pi\big(T \cdot (\pi^{-1}(\boldsymbol{u}_i) \cdot d)\big)$



[Jin,Favaro,Soatto'03] [Silveira, Malis, Rives, TRO'08], [Newcombe et al., ICCV '11], [Engel et al., ECCV'14], [Forster et al., ICRA'14]

Indirect vs Direct method



https://cse.sc.edu/~yiannisr/774/2015/eccv2014.pdf

Indirect vs Direct method

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)

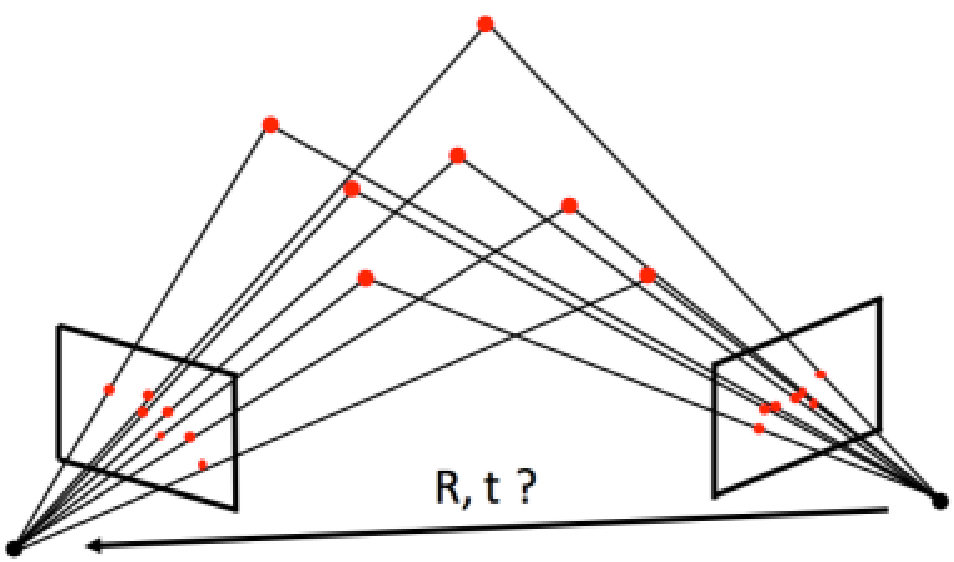
- How to process image info?
- How to select interesting points on image frame?
- How do we find and use the connection between consecutive frames?
- How to extract motion from frames?
- Does every frame should be treated equally?
- What is front end and back end?
- Why do we need initialization?

…

Visual processing pipeline

VIO/SLAM is mainly divided into two parts: the **front end** and the **back end**. **Front end** roughly estimates the motion of adjacent images as well as IMU preintegration constraint and provides a **good initial value for the back end**.

Front End $\xrightarrow{\text{initialization}}$ Back End



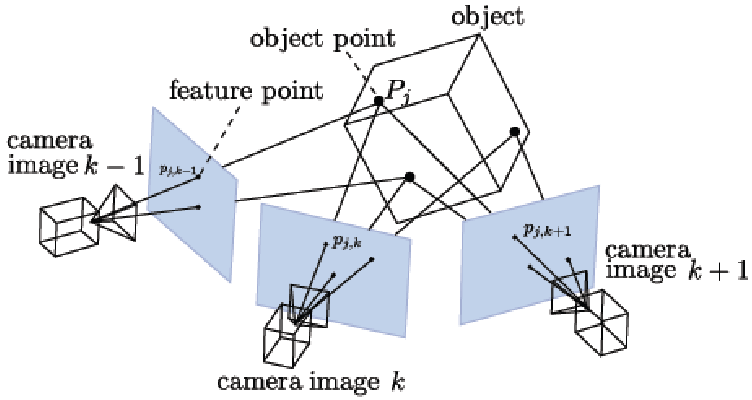
1. Data Selection
    1. Point
    2. Frame
2. Data Association
    1. Feature Matching
    2. Optical Flow
3. Initial Pose and Depth Estimation
    1. Geometric approach
    2. Direct Image alignment
    3. Pose-only BA

Data Selection --- Geometry

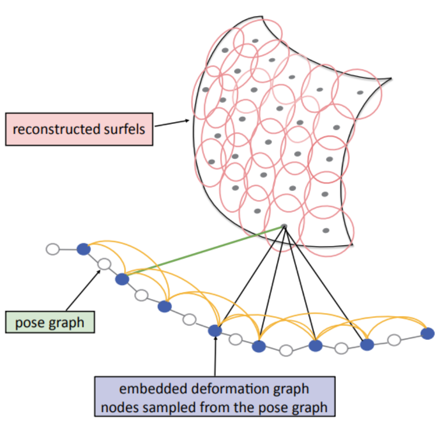
Geometry

Method	Sensors	Back-end	Geometry
ORB-SLAM [22]	mono	g2o	points
DSO [23]	mono	g2o	points
VINS-mono [24]	mono/IMU	Ceres	points
VINS-Fusion [25]	mono/Stereo/IMU	Ceres	points
ROVIOLI [26]	stereo/IMU	EKF	points
ElasticFusion [18]	RGB-D	alternation	surfels
Voxblox [27]	RGB-D	[26]	TSDF

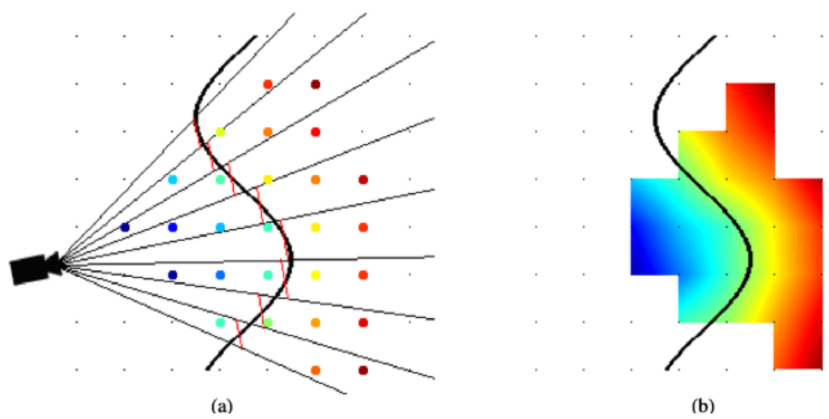
open-source libraries for visual and visual inertial SLAM



Points



surfel



truncated signed distance function (**TSDF**)

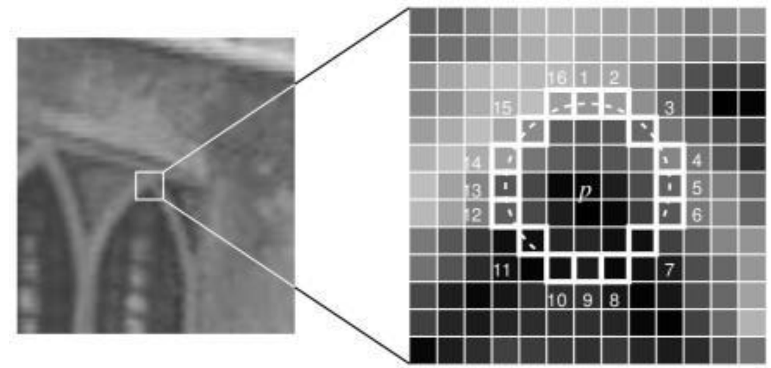
Gomez-Ojeda, et al. "PL-SLAM: a stereo SLAM system through the combination of points and line segments." *IEEE Transactions on Robotics* 35.3 (2019).
Fu, Xingyin, et al. "Real-time large-scale dense mapping with surfels." *Sensors* 18.5 (2018): 1493.

Data Selection --- Geometry

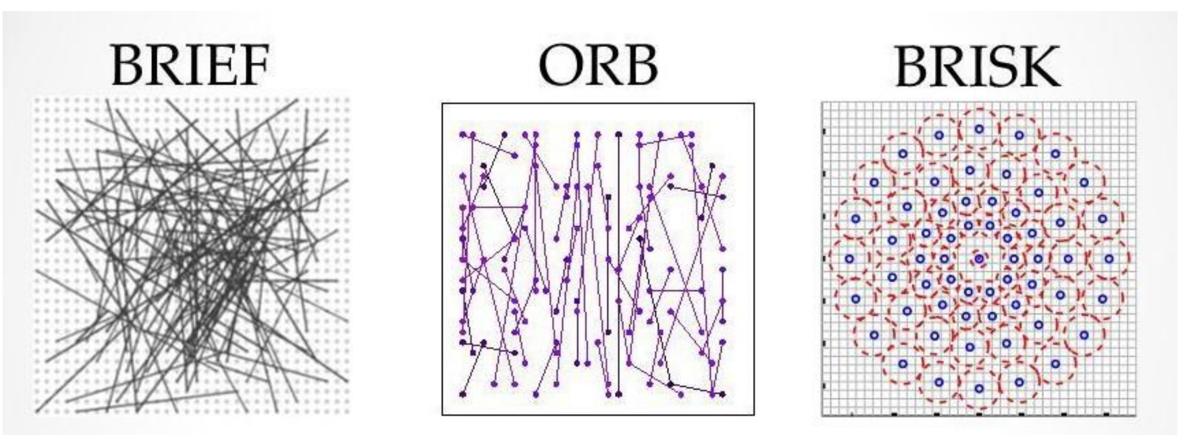
NUS National University of Singapore

FAST Corner Detection: points with weak intensity variations

The pixel p is a corner if there exists a set of n contiguous pixels in the circle (of 16 pixels) which are all brighter than Ip+t, or all darker than Ip−t.

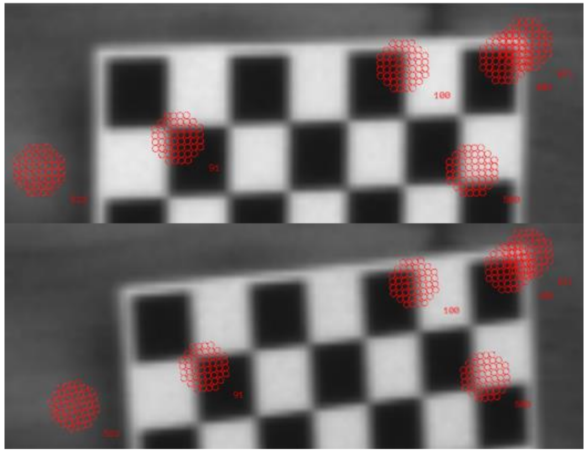


Indirect Method



Feature descriptor (fingerprint)

Direct Method



Patch around the feature

Data Selection --- frame

Keyframe: the sub-set of frames we selected to do successive refinement steps which usually applied by **iterative non-linear optimization** techniques—such as bundle adjustment.

Typical selection criteria (from the last keyframe to the latest frame)

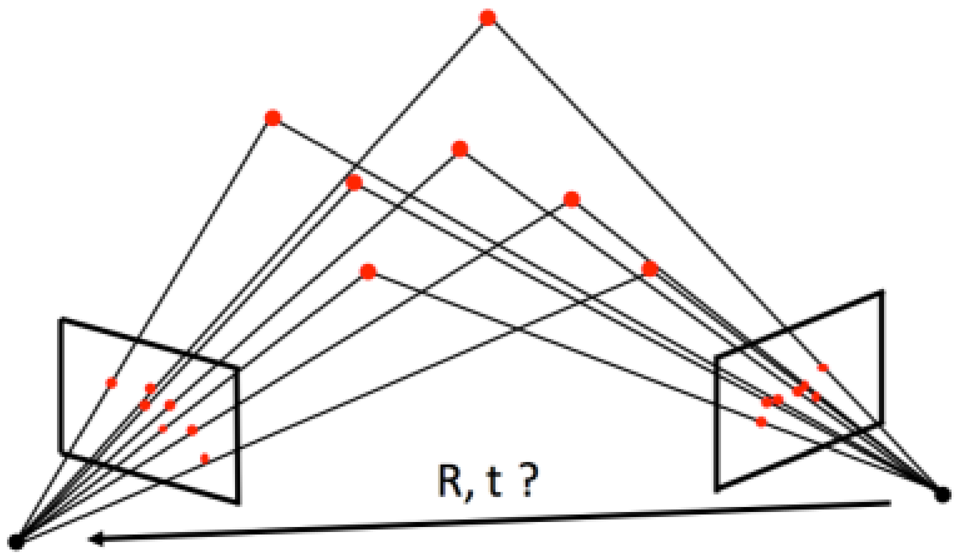
- Pose change bigger than certain threshold
- Mean square optical flow larger than certain threshold

during initial coarse tracking.

$$f := \left(\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{p}-\mathbf{p}'\|^2\right)^{\frac{1}{2}}$$

- Photometric difference bigger than certain value
- …

Front End $\xrightarrow{\text{initialization}}$ Back End

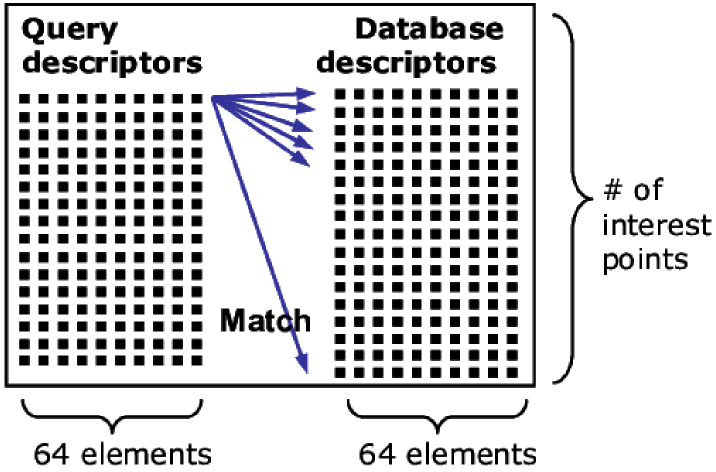


1. Data Selection
    1. Point
    2. Frame
2. Data Association
    1. Feature Matching
    2. Optical Flow
3. Initial Pose and Depth Estimation
    1. Geometric approach
    2. Direct Image alignment
    3. Pose-only BA

Data association
Indirect method (feature matching)

Algorithms

- Brute-Force Matcher
- FLANN(Fast Library for Approximate Nearest Neighbors) Matcher



How do we improve this **time consuming** feature matching module in indirect method ?

Use optical flow!

Data association
Indirect method (Optical Flow)

Optical Flow: Given two consecutive image frames, estimate the **motion of each pixel**

Assumptions: **Brightness constancy** and **Small motion**

Intensity function $I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$

Linearize it with multivariable Taylor series expansion

$$I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x, y, t)$$

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

flow velocities

$$I_x u + I_y v + I_t = 0$$

Image gradients
(at a point p)

temporal gradient

http://www.cs.cmu.edu/~16385/lectures/lecture24.pdf

Example of image and temporal gradients

flow velocities

$$I_x u + I_y v + I_t = 0$$

Image gradients
(at a point p)

temporal gradient

3 x 3 patch

t

1	1	1	1	1
1	1	1	1	1
1	**10**	**10**	**10**	**10**
1	**10**	**10**	**10**	**10**
1	**10**	**10**	**10**	**10**
1	**10**	**10**	**10**	**10**

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	**10**	**10**	**10**
1	1	**10**	**10**	**10**
1	1	**10**	**10**	**10**

$$I_x = \frac{\partial I}{\partial x}$$

-	0	0	0	-
-	0	0	0	-
-	**9**	0	0	-
-	**9**	0	0	-
-	**9**	0	0	-
-	**9**	0	0	-

-1 0 1

$$I_y = \frac{\partial I}{\partial y}$$

-	-	-	-	-
0	0	0	0	0
0	**9**	**9**	**9**	**9**
-	0	0	0	0
0	0	0	0	0
-	-	-	-	-

-1
0
1

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	**9**	**9**	**9**	**9**
0	**9**	0	0	0
0	**9**	0	0	0
0	**9**	0	0	0

http://www.cs.cmu.edu/~16385/lectures/lecture24.pdf

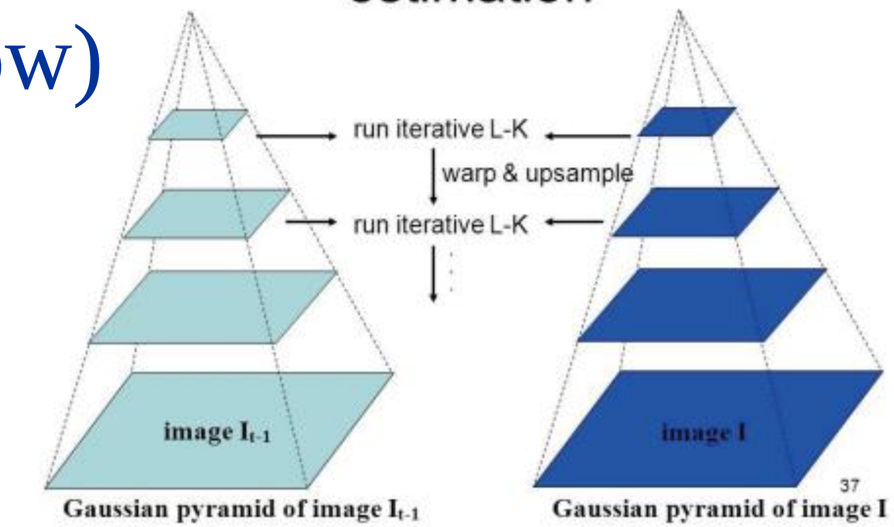
Data association
Indirect method (Optical Flow)

unknown

$$I_x u + I_y v + I_t = 0$$

known

Coarse-to-fine optical flow estimation



Using a 5 x 5 image patch, gives us 25 equations

$$\begin{aligned}
I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v &= -I_t(\boldsymbol{p}_1)\\
I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v &= -I_t(\boldsymbol{p}_2)\\
&\vdots\\
I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v &= -I_t(\boldsymbol{p}_{25})
\end{aligned}$$

$$\underbrace{\begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix}}_{A^\top A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\hat{x}} = - \underbrace{\begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix}}_{A^\top b}$$

$$x = (A^\top A)^{-1} A^\top b$$

Use optical flow results as **initial guess** for feature matching

http://www.cs.cmu.edu/~16385/lectures/lecture24.pdf

Data association --- Direct method

Direct minimization of photometric error

$$E(\xi) = \sum_{\mathbf{p}_i \in \Omega_{\text{ref}}} \left(I_{\text{ref}}(\mathbf{p}_i) - I(\mathbf{p}'_i) \right)^2$$

camera pose

sum over valid ref. pixel

ref. image

new image

ref. depth

$$\mathbf{p}'_i = \omega(\mathbf{p}_i, d, \xi) = \pi(K(R_\xi K^{-1} d \begin{pmatrix} \mathbf{p}_{i,x} \\ \mathbf{p}_{i,y} \\ 1 \end{pmatrix} + \mathbf{t}_\xi))$$

$$\pi(x, y, z) := \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

$$\begin{pmatrix} R_\xi & \mathbf{t}_\xi \\ \mathbf{0} & 1 \end{pmatrix} := \exp(\widehat{\xi})$$

$\omega(\mathbf{p}_i, d, \xi)$ „warps" a pixel from ref. image to new image

http://www.dis.uniroma1.it/~labrococo/tutorial_icra_2016/icra16_slam_tutorial_engel.pdf

Data association --- Direct method

Iterative the following steps till converge to solve the photometric optimization problem

$$E(\xi) = \sum_{\mathbf{p}_i \in \Omega_{\text{ref}}} \left(I_{\text{ref}}(\mathbf{p}_i) - I(\mathbf{p}_i') \right)^2$$

1. Linearize wrt. left-multiplied increment to pose $\xi^{(n)}$:

$$J_{\mathbf{r}} := \left. \frac{\partial \mathbf{r}(\epsilon \circ \xi^{(k)})}{\partial \epsilon} \right|_{\epsilon=0} \in R^{n \times 6}$$

2. Solve for increment

$$\delta_\xi = -(J_{\mathbf{r}}^T J_{\mathbf{r}})^{-1} J_{\mathbf{r}}^T \mathbf{r}(\xi^{(k)}))$$

3. Apply increment $\xi^{(k+1)} = \delta_\xi \circ \xi^{(k)}$

http://www.dis.uniroma1.it/~labrococo/tutorial_icra_2016/icra16_slam_tutorial_engel.pdf

Data association

Relationship between optical flow and direct method

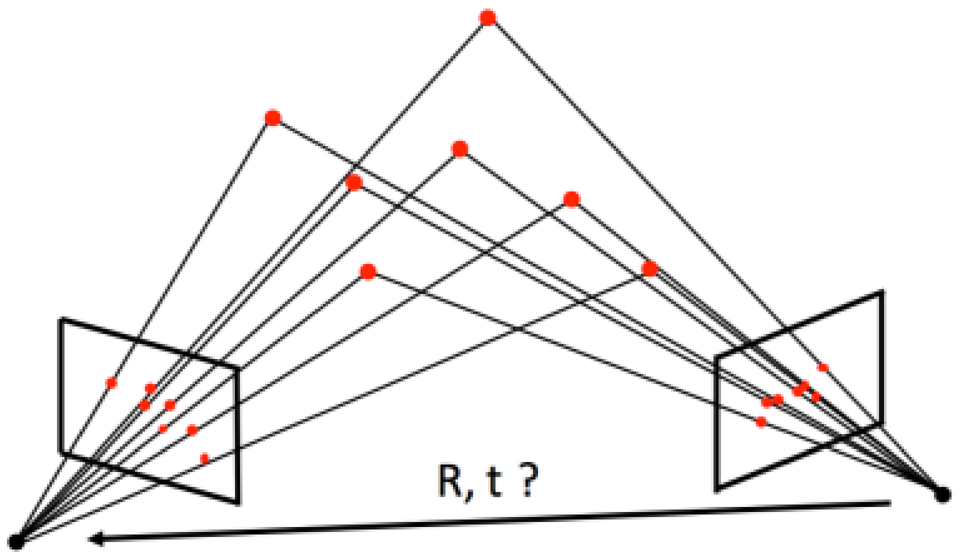
Direct method derived from optical method

- Both have strong assume on **brightness consistent** (not suitable for strong reflection scenario, e.g. metal and glass)

Differences:

- Optical flow normally linearizes the intensity function **wrt. pixel coordinate**. (It could be generalized to apply with warp function)
- Direct method linearizes the cost function **wrt. 6D pose** parameter
- Direct method satisfies implicitly the **epipolar constrain**, while optical flow violates the epipolar constraints

Front End $\xrightarrow{\text{initialization}}$ Back End



1. Data Selection
    1. Point
    2. Frame
2. Data Association
    1. Feature Matching
    2. Optical Flow
3. Initial Pose and Depth Estimation
    1. Geometric approach
    2. Direct Image alignment
    3. Pose-only BA

Initial pose and depth estimation

Initialization of pose and points at the very beginning:

Matched **points set or flow** → Retrieve **pose** from F or H matrix → Triangulation to get 3d map points or points depth relative to certain frame

Tracking after the system has already initialized:

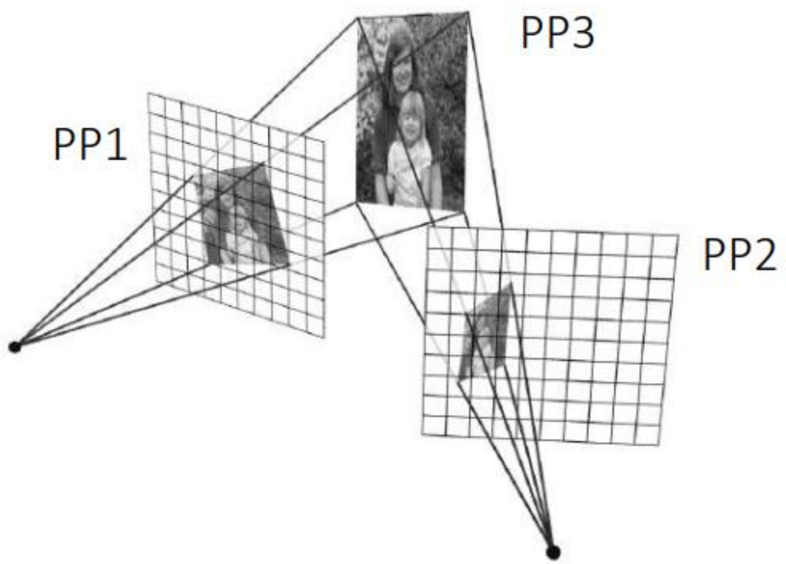
Project map points to current frame → Solve pose
- Indirect: pose only Bundle Adjustment
- Direct: image alignment

→ Obtain 3D points or depth if necessary

What is Essential, Fundamental, and Homography matrix?
How to do triangulation to get 3D points?

Initial pose and depth estimation

Homograph (H) Matrix



A projective transformation a.k.a. a Homograph (H) Matrix is the kind of transformation to warp projective plane 1 into projective plane 2

$$P' = H \cdot P \quad \text{or} \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

When can we use homographies?

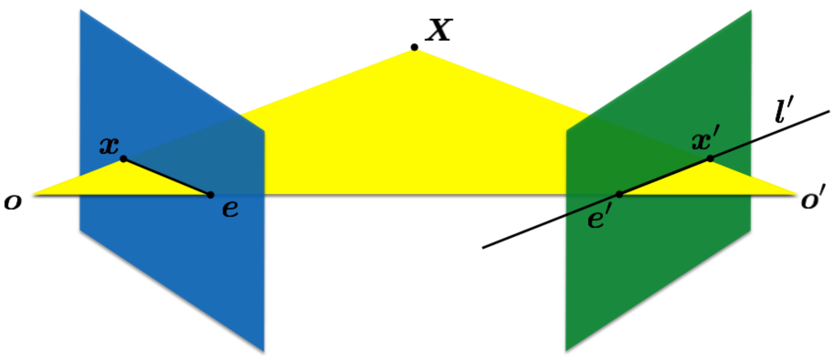
1. the scene is planar;

2. the scene is very far or has small (relative) depth variation → scene is approximately planar

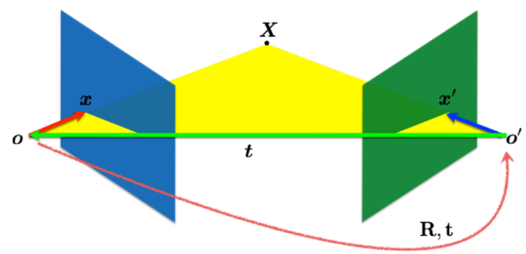
http://www.cs.cmu.edu/~16385/lectures/lecture9.pdf

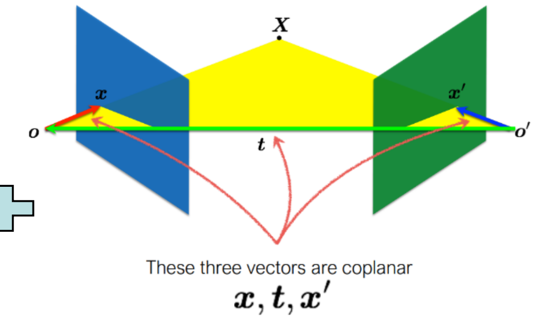
Initial pose and depth estimation

Essential (E) Matrix



$$\boxed{x'^{\top}\mathbf{E}x = 0}$$





rigid motion

$$x' = \mathbf{R}(x - t)$$

coplanarity

$$(x - t)^{\top}(t \times x) = 0$$

$$(x'^{\top}\mathbf{R})(t \times x) = 0$$

$$(x'^{\top}\mathbf{R})([\mathbf{t}_{\times}]x) = 0$$

$$x'^{\top}(\mathbf{R}[\mathbf{t}_{\times}])x = 0$$

$$x'^{\top}\mathbf{E}x = 0$$

The fundamental matrix is a generalization of the essential matrix, where the assumption of Identity matrices is removed

$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

http://www.cs.cmu.edu/~16385/lectures/lecture12.pdf

Initial pose and depth estimation

How to solve F, E, or H matrix?

Assume we have M matched image points

$$\{\boldsymbol{x}_m, \boldsymbol{x}'_m\} \qquad m = 1, \ldots, M$$

Each correspondence should satisfy

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0 \;\text{ or }\; \boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$$

Then with at least 5 points you can solve for the 3x3 E matrix and with at least 4 points pair the 3x3 H matrix could be solved.

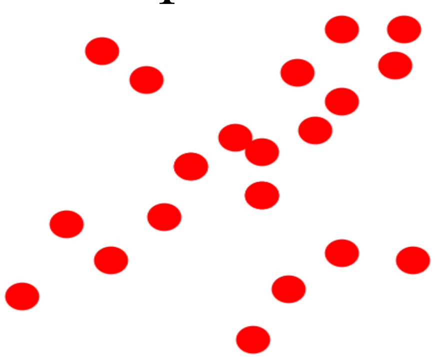
Problem	Inputs	Model to find	Basic Equation	d.o.f.	Min. # of matches	Minimal solution
Camera Location	$\mathbf{u}_{ij}, \mathbf{x}_{wj}$	Pose $\mathbf{T}_{iw}$	$\pi_i(\mathbf{T}_{iw}, \mathbf{x}_{wj})$	6	3	p3p
Initialize 3D scene	$\mathbf{u}_{1j}, \mathbf{u}_{2j}$	Essential Matrix $\mathbf{E}_{12} = [\mathbf{t}]_\times \mathbf{R}$	$\mathbf{u}_{1j}^T \mathbf{E}_{12} \mathbf{u}_{2j} = 0$	5	5	5-point 8-point
Initialize 2D scene	$\mathbf{u}_{1j}, \mathbf{u}_{2j}$	Homography $\mathbf{H}_{12}$	$\mathbf{u}_{1j} = \mathbf{H}_{12}\mathbf{u}_{2j}$	8	4	

http://www.dis.uniroma1.it/~labrococo/tutorial_icra_2016/icra16_slam_tutorial_tardos.pdf

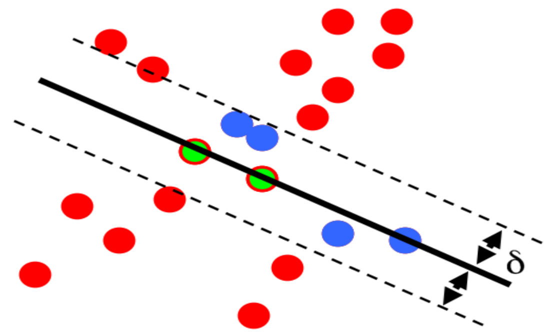
Initial pose and depth estimation

RANSAC : Find matching points that agree with the H or F matrix

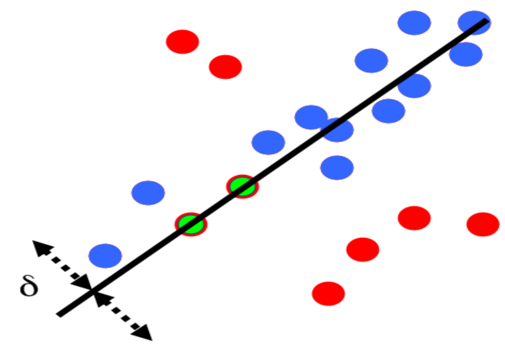
Example 1: Fitting lines with outliers



Data points

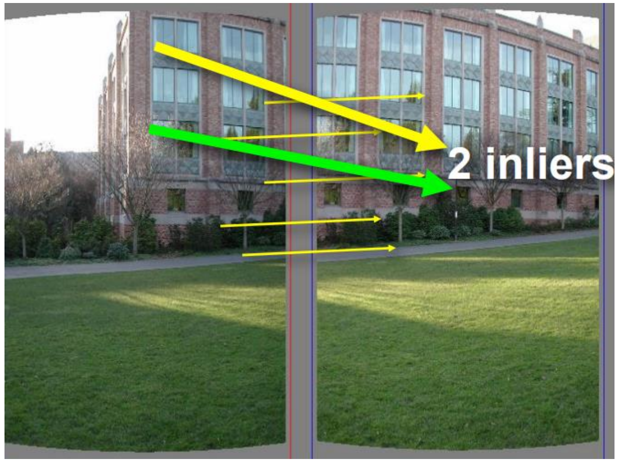


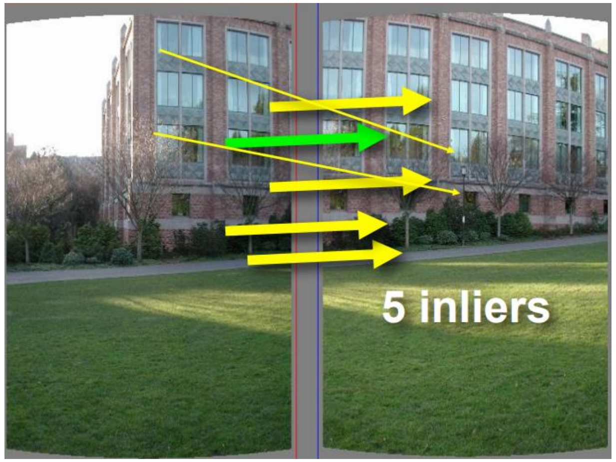
Inline count: N = 6



N = 14

Example 2





Initial pose and depth estimation

Search for consensus with a robust technique: RANSAC

```
RANSAC (P) return M and S
-- P: set of potential matches
-- M: alignment model found (requires at least k matchings)
-- S: set of supporting matches
for i = 1..max_attempts
    Si ← choose randomly k matchings from P
    Mi ← compute alignment model from Si
    Si* ← matchings in P that agree with Mi  (with tolerance ε )
    if  #(Si*) > consensus_threshold
        Mi* ← compute alignment model from Si*  (using least squares)
        return Mi* and Si*
    end if
endfor
return failure
```

Initial pose and depth estimation

Model selection in initialization: **Essential Matrix vs Homography**

They are both 3 x 3 matrices but …

$$l' = \mathbf{E}x \qquad\qquad x' = \mathbf{H}x$$

Essential matrix maps a **point** to a **line**

Homography maps a **point** to a **point**

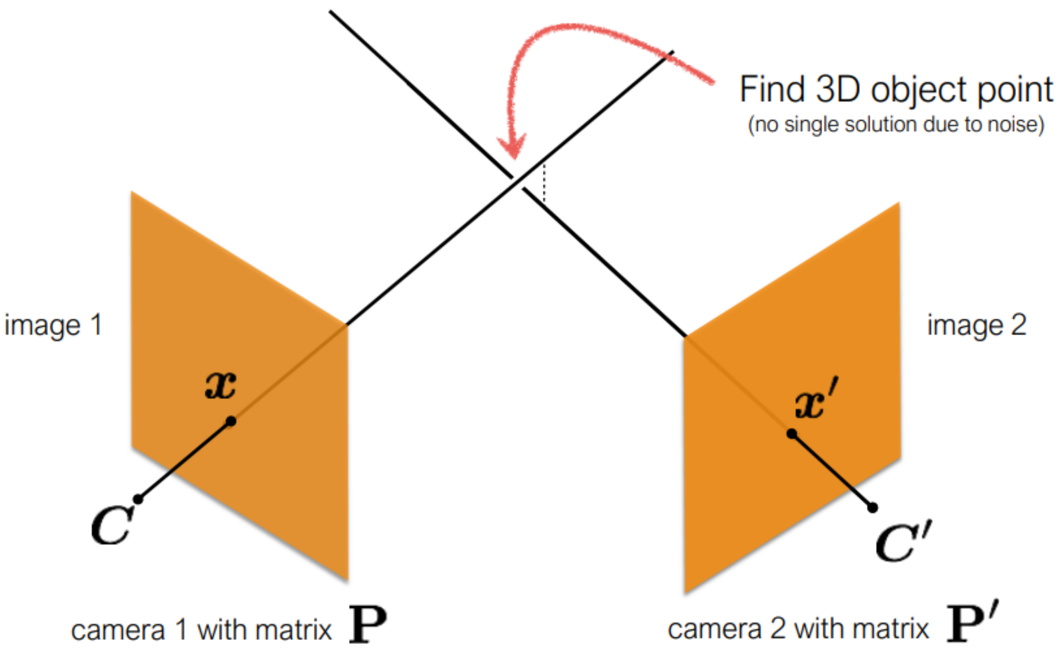
	H matrix	F matrix
Scene	A (nearly) **planar scene** or when there is low parallax	A **non-planar scene** with enough parallax
Retrieved motion hypotheses	8	4

Get the best solution with most points seen in front of both cameras and with low Reprojection error from motion hypotheses

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, *31*(5), 1147-1163.

Initial pose and depth estimation

Triangulation



Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

Estimate the 3D point

$$\mathbf{X}$$

$$\mathbf{x} = \alpha \mathbf{P} \boldsymbol{X}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

http://www.cs.cmu.edu/~16385/lectures/lecture12.pdf

Back End

Front End $\xrightarrow{\text{initialization}}$ Back End

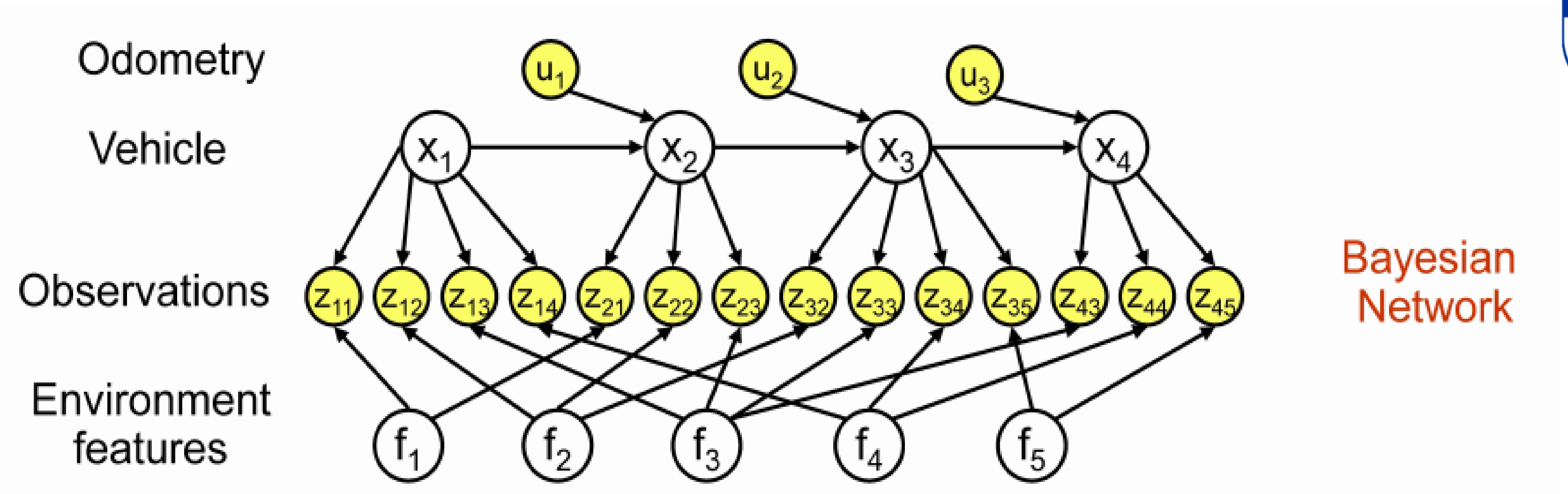
Minimize a non-linear energy that consists of reprojection terms, IMU terms

$$E = \boxed{\sum_{\substack{i \in \mathcal{P} \\ t \in \mathrm{obs}(i)}} \mathbf{r}_{it}^{\top} \mathbf{\Sigma}_{it}^{-1} \mathbf{r}_{it}} + \boxed{\sum_{(i,j) \in \mathcal{C}} \mathbf{r}_{ij}^{\top} \mathbf{\Sigma}_{ij}^{-1} \mathbf{r}_{ij}} \, .$$

Reprojection/Photometric terms are summed over the set of points P and for each point i over the set obs(i) of frames where the point is observed

IMU preintegration factors summed over the set C which contains pairs of frames which are connected by IMU constraint

Solve the **non-linear least-squares optimization problem** with Gaussian-Newton or Levenberg-Marquardt



How many frames/nodes/states do we need to consider during the back end optimization?

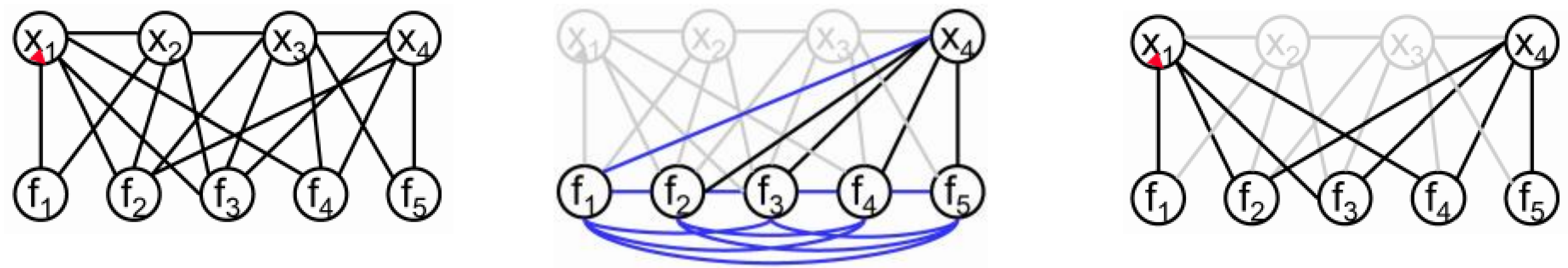
Highly correlated with the computational demand and accuracy

http://www.dis.uniroma1.it/~labrococo/tutorial_icra_2016/icra16_slam_tutorial_tardos.pdf

Three major tightly coupled VIO categories

Categorized by the number of camera-poses involved in the estimation :

- Filtering methods only estimate the latest state.
- Full state optimization (or batch nonlinear least-squares algorithms) optimize the complete history of states
- Fixed-lag optimization (or sliding window estimators) consider a window of the latest states



Original Problem | Filter approach | keyframe optimization method

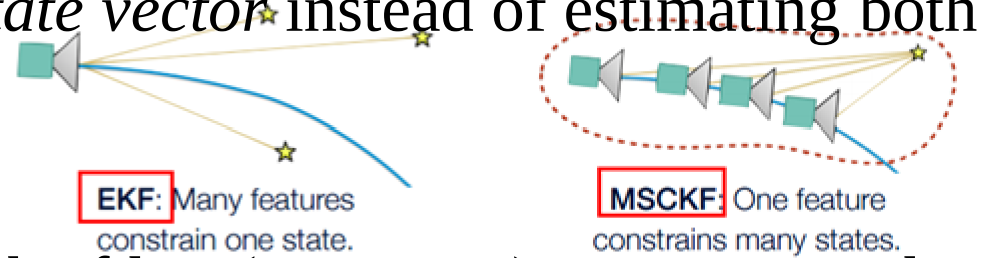
http://www.dis.uniroma1.it/~labrococo/tutorial_icra_2016/icra16_slam_tutorial_tardos.pdf

Filtering algorithms

Filtering algorithms enable efficient estimation by restricting the inference process to the latest state of the system.

Typical work: Multi-State Constraint Kalman filter (MSCKF)

A structure-less approach where *landmark positions are marginalized out of the state vector* instead of estimating both the poses and landmarks



Pros

Avoid the complexity of the filter (e.g., EKF) growing quadratically in the number of estimated landmarks.

Cons

Less accuracy: the processing of landmark measurements needs to be delayed until all measurements of a landmark are obtained

Mourikis, Anastasios I., and Stergios I. Roumeliotis. "A multi-state constraint Kalman filter for vision-aided inertial navigation." *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007.

Full state optimization

Full smoothing methods estimate the entire history of the states by solving a large nonlinear optimization problem

Pros: guarantees the **highest accuracy**, since it update the linearization point of the complete state history as the estimate evolves.

Cons: the **complexity** of the optimization problem is approximately cubic with respect to the dimension of the states

Common practice:

- keep selected keyframes (ORB SLAM)
- run optimization in a parallel tracking and mapping architecture (SVO)
- incremental smoothing techniques (iSAM2)

Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system." *IEEE transactions on robotics* 31.5 (2015): 1147-1163.
Forster, Christian, et al. "SVO: Semidirect visual odometry for monocular and multicamera systems." *IEEE Transactions on Robotics* 33.2 (2016): 249-265.
Kaess, Michael, et al. "iSAM2: Incremental smoothing and mapping using the Bayes tree." *The International Journal of Robotics Research* 31.2 (2012): 216-235.

Fixed-lag Optimization

Fixed-lag smoothers estimate the **states** that fall within a given **time window**, while marginalizing out older states.

Pros:

- more **accurate** than filtering

Cons:

- the marginalization of the states outside the estimation window can lead to dense Gaussian priors, which hinders efficient matrix operations. (Can be solved with factor recovery method etc.)

Typical work:

Basalt: Visual-Inertial Mapping with Non-Linear Factor Recovery

Usenko, Vladyslav, et al. "Visual-inertial mapping with non-linear factor recovery." *IEEE Robotics and Automation Letters* (2019).

Framework Example: SVO

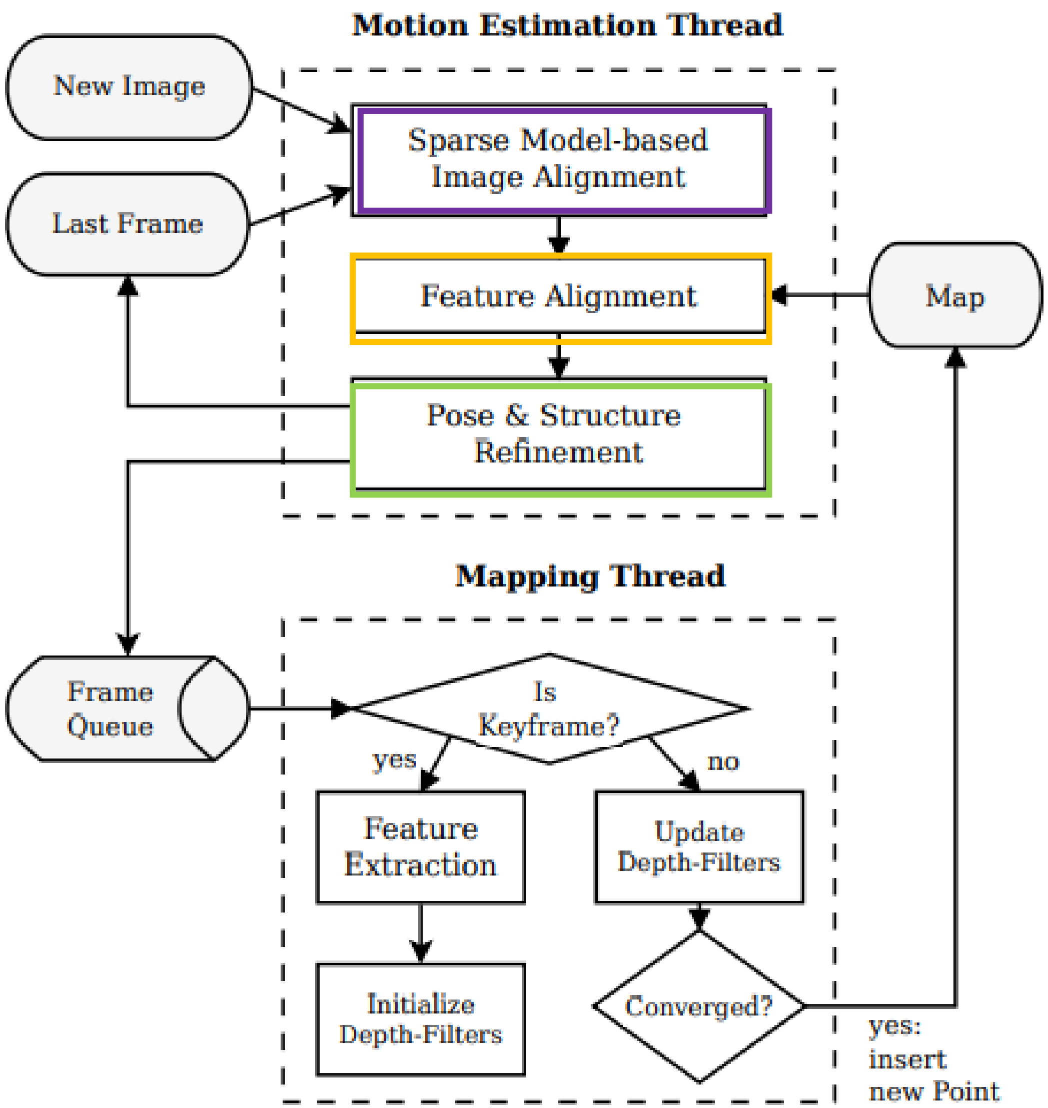
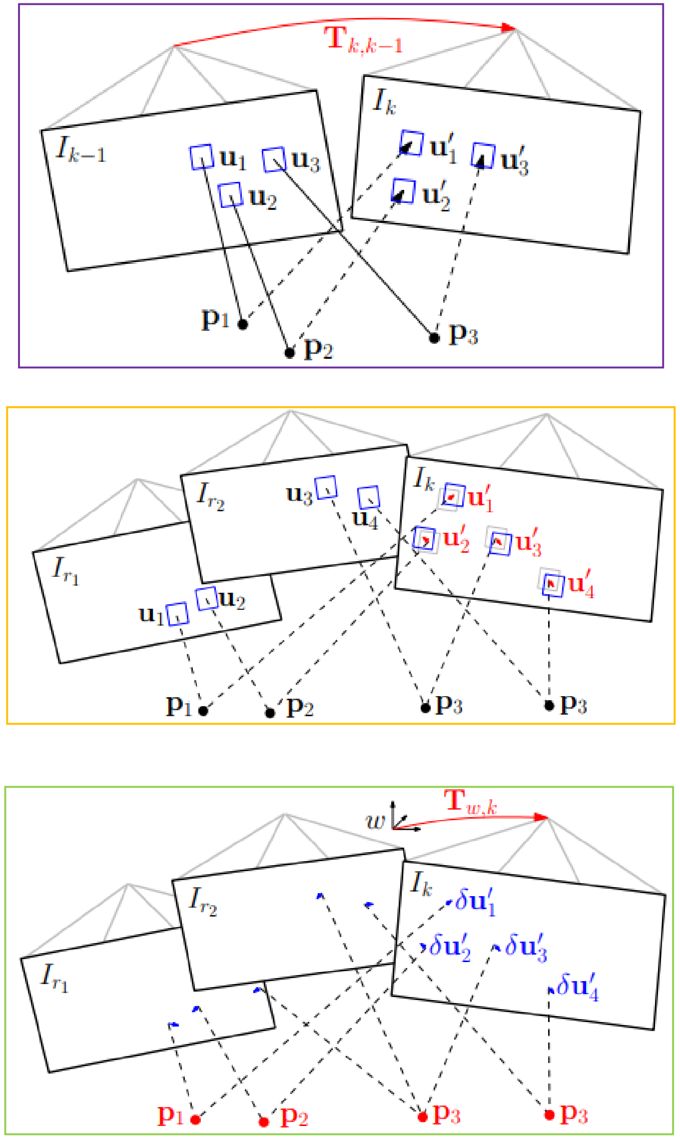


Fig. 1: Tracking and mapping pipeline



red: parameters to optimize
blue: optimization cost

https://www.cnblogs.com/luyb/p/5773691.html

Our next plan regarding VIO

VO Front End Improvement

- IMU prior integration, for robust feature tracking under high rotational motion

Computational Cost Reduction

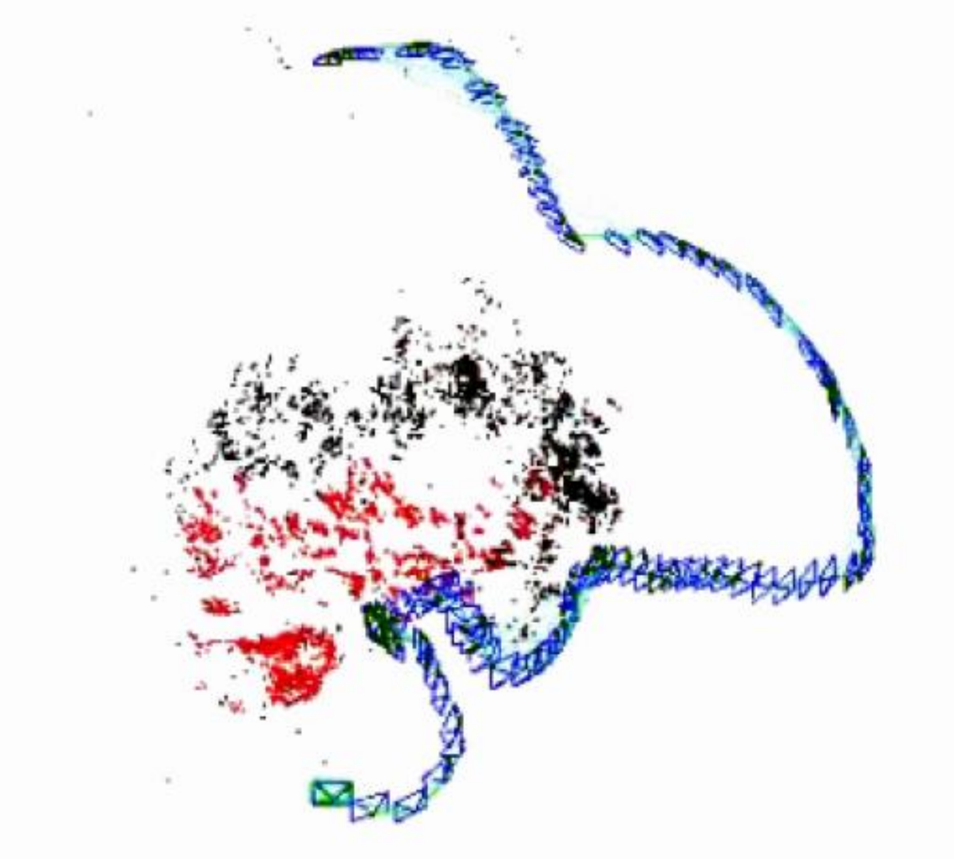
- Visual Odometry computation with known depth generated by simulator and Pengfei's Algorithm, removing triangulation calculation in mapping.

Summary

- SLAM and VIO problem formulation
- Observation model: Geometry/Photometric measurement
- Front End in SLAM/VIO
    - Direct and indirect method
    - Optical flow
    - Data selection and association
    - Visual initialization
        - Basics in homography, epipolar geometry and triangulation
- Common practice of SLAM/VIO: filtering and optimization based
- Our short-term plan

Key topics not covered here

- Lie Group and rigid body Kinematics
- IMU Initialization in VIO
- IMU preintegration details
- Depth filter
- Back end optimization
- Loop closure
- Fisheye camera model
- Deep Learning Adaption
    - FlowNet
    - MonoDepth
    - …



Thank you！

Zhou Yu