Shanghai Jiao Tong University

StructSLAM: Visual SLAM with Building Structure Lines

Danping Zou

Assistant Professor

Key Laboratory of Navigation and Location-based Services

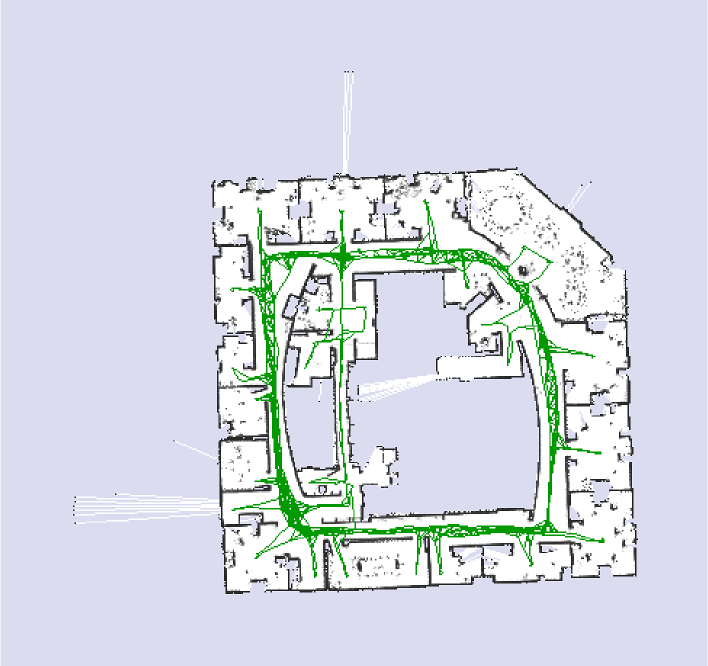
Shanghai Jiao Tong University

dpzou@sjtu.edu.cn

http://drone.sjtu.edu.cn/dpzou

What's SLAM?

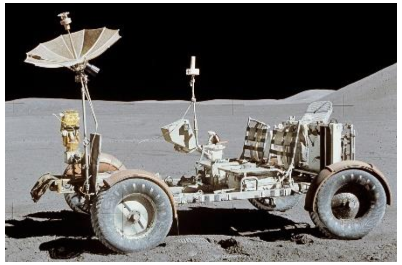
- SLAM :**S**imultaneous **L**ocalization **A**nd **M**apping.



Constructing a map of an unknown environment while simultaneously keeping track of the robot's location.

- Autonomous navigation
- motion planning

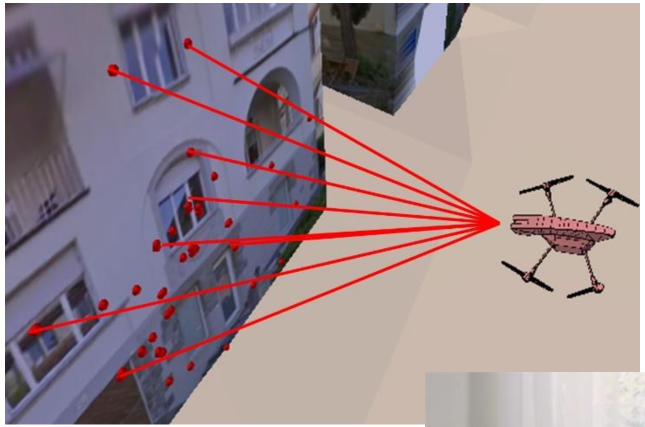





Active areas for SLAM research

Robotics

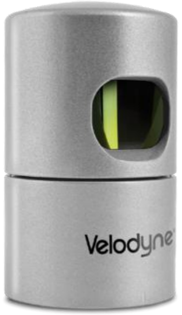
- UAVs， Service robots



Augmented Reality (AR)

SLAM with different kinds of sensors

2D laser rangefinder



3D liDar

RGB-D camera

- Active sensing
- Heavy and bulky
- Active sensing
- Energy consuming

Since 2005, there has been intense research into VSLAM (Visual SLAM) using primarily visual (camera) sensors.

- Passive sensing
- Light & compact
- Energy saving
- Ubiquity

List of Visual SLAM methods

- MonoSLAM
- FastSLAM
- GraphSLAM
- PTAM
- ORB-SLAM
- **CoSLAM***
- DTAM
- LSD-SLAM

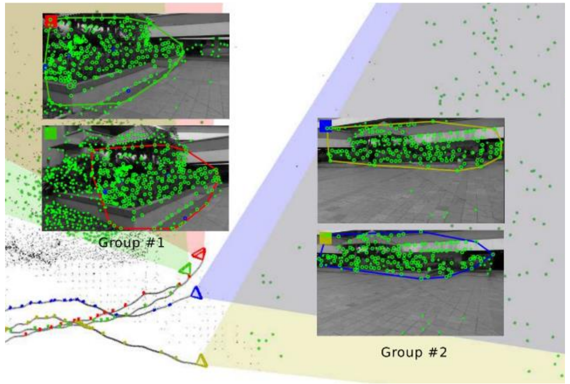
Extended Kalman filter

Structure-from-motion

Sparse feature points

Dense / Semi dense

*Zou, Danping, and Ping Tan. "CoSLAM: Collaborative visual slam in dynamic environments." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.2 (2013): 354-366.



Knowing issues for Visual SLAM

- Scale ambiguity
- Lack of Illumination
- Accumulated error (Drift error)
- Lack of Texture

Vision + X

GNSS (GPS、Beidou)

Inertial Units (Gyro+Acc+Compass)

Wireless（Wifi , Bluetooth)

Map / Floor plan

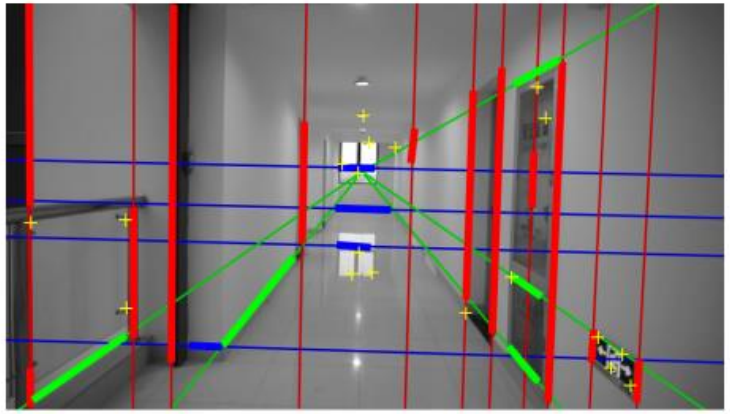
What about if we consider one special case : indoor scenes

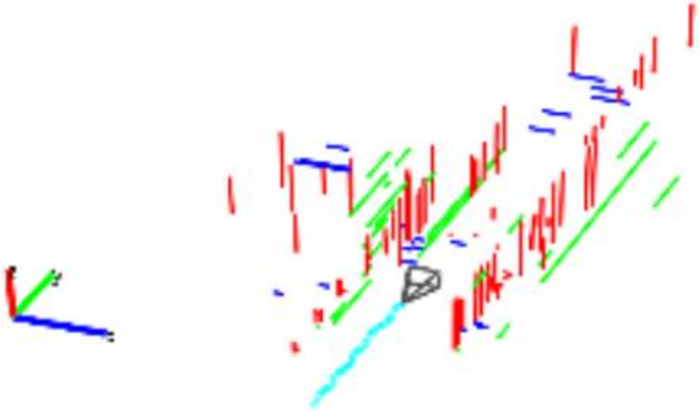
Natural scenes

Man-made scenes

StructSLAM

- A novel visual SLAM method aided with structure lines
  - Structure lines : The line feature aligned with dominant orientations





Zhou, Huizhong, Danping, Zou, et al. "StructSLAM: Visual SLAM with building structure lines." *Vehicular Technology, IEEE Transactions on* 64.4 (2015): 1364-1375.
- Special session for indoor localization

Indoor scenes

Motivations:

- Lines are better landmarks in texture-less scenes (like many indoor scenes with only white walls) than points.
- Some lines (structure lines) encode the global orientation information.

$\underbrace{\hspace{10em}}$

Less accumulated error

StructSLAM:

Visual SLAM with Building Structure Lines

Hui Zhong Zhou, Danping Zou et al.

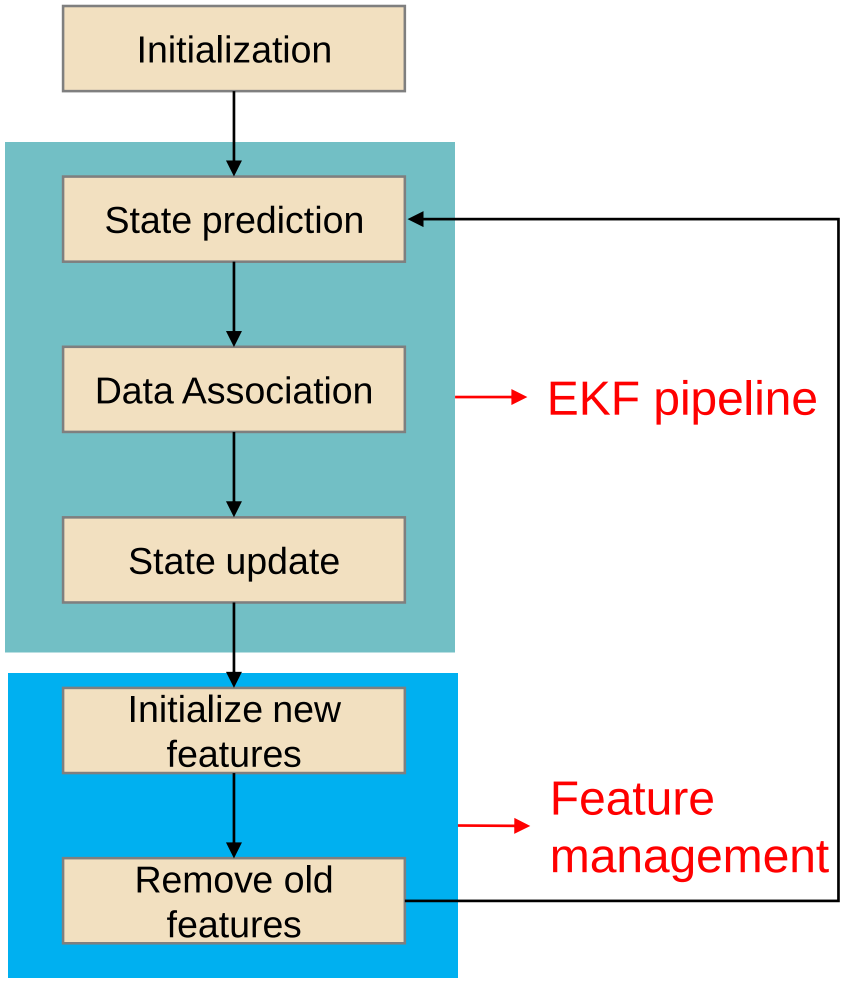
Shanghai Key Laboratory of Navigation and Location Based Services
Shanghai Jiao Tong University
Apirl,2014

http://drone.sjtu.edu.cn/dpzou/project/structslam.php

Pipeline of StructSLAM

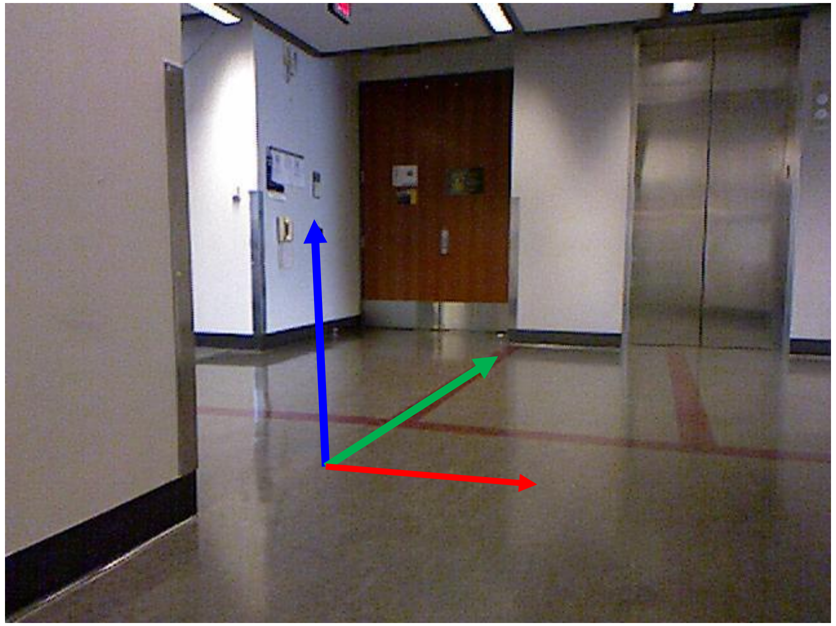
Extended Kalman filter:

- Simple to be implemented
- Easily fused with other sensors (IMU, odometers)
- More robust than structure-from-motion pipeline.



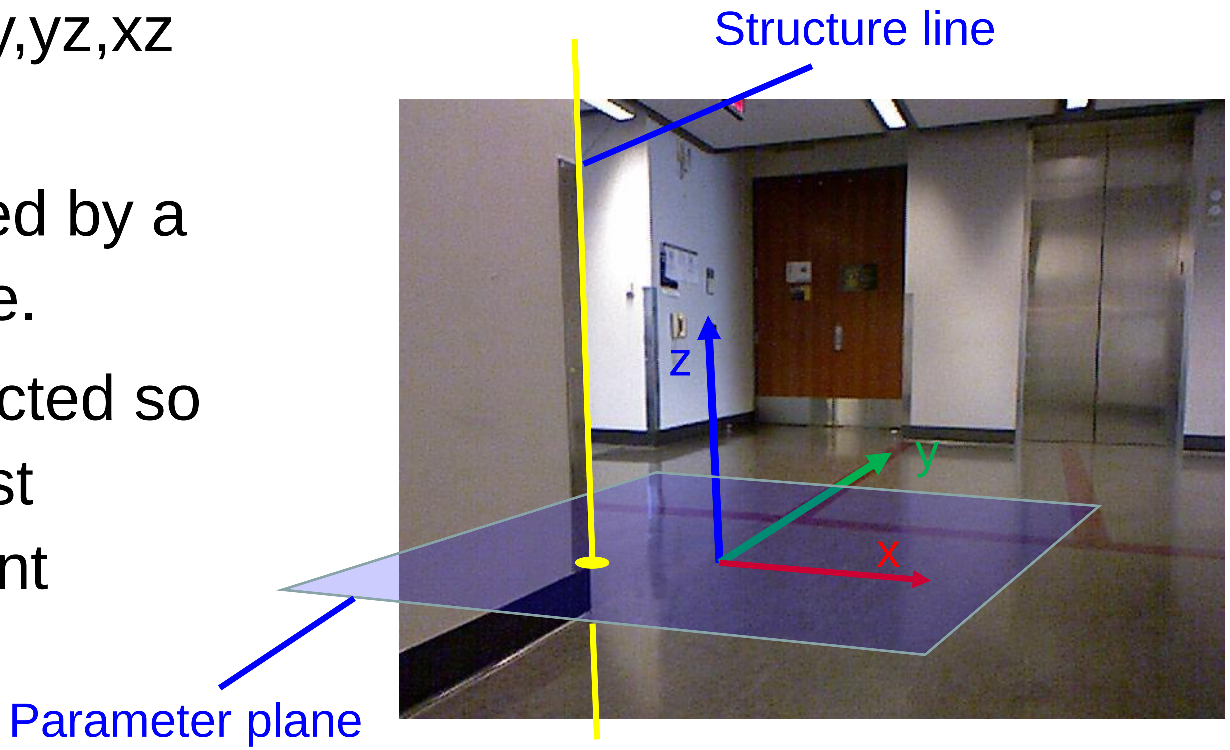
Dominant directions

- The man-made world are generally dominated by several directions.
  - Vertical direction (always points to the sky)
  - Horizontal directions (are usually perpendicular to each other, although not always)



Parameter planes

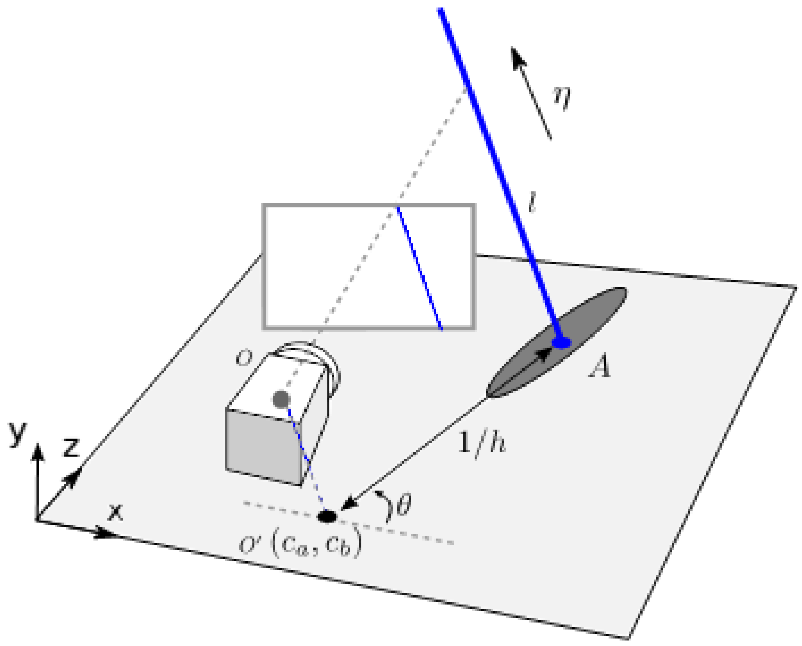
- Parameter plane is one of xy,yz,xz planes of the world frame.
- A structure line is represented by a point on the parameter plane.
- The parameter plane is selected so as to make sure it is the most perpendicular to the dominant direction.



Structure line

- Each structure line is represented by a point on the parameter plane, denoted by a 4x1 vector.
- It is in fact a 2D inverse depth representation*

$$\mathbf{l} = \begin{pmatrix} c_a \\ c_b \\ \theta \\ h \end{pmatrix} \begin{array}{l} \left.\vphantom{\begin{matrix} c_a \\ c_b \end{matrix}}\right\} \text{Projection of camera center} \\ \} \text{Direction} \\ \} \text{Inverse depth} \end{array}$$



* Montiel, J. M. M., Javier Civera, and Andrew J. Davison. "Unified inverse depth parametrization for monocular SLAM." *analysis* 9 (2006): 1.

StructSLAM – State representation

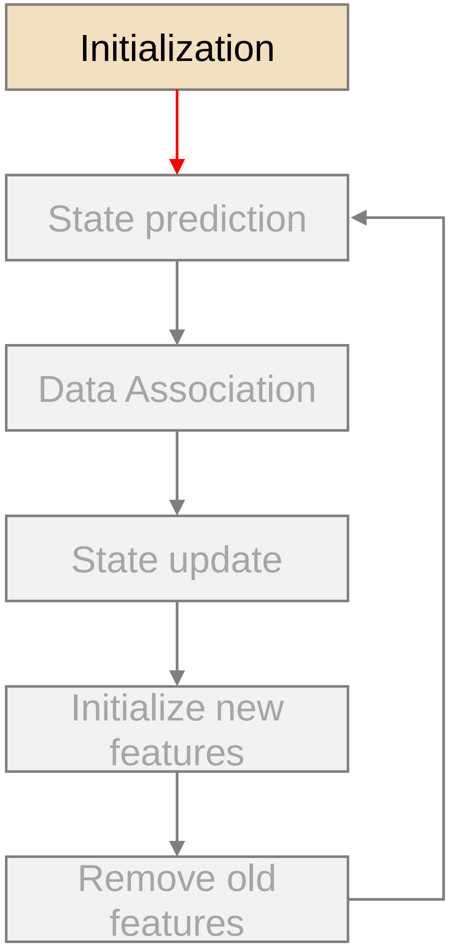
- State vector and covariance matrix (MonoSLAM)

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_p \\ \boxed{\mathbf{x}_l} \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{cc} & \boldsymbol{\Sigma}_{cp} & \boldsymbol{\Sigma}_{cl} \\ \boldsymbol{\Sigma}_{pc} & \boldsymbol{\Sigma}_{pp} & \boldsymbol{\Sigma}_{pl} \\ \boldsymbol{\Sigma}_{lc} & \boldsymbol{\Sigma}_{lp} & \boldsymbol{\Sigma}_{ll} \end{bmatrix}$$

Camera pose + Points + Structure Lines

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{p}^w \\ \mathbf{q}^{wc} \\ \mathbf{v}^w \\ \omega^c \end{bmatrix} \qquad \mathbf{x}_p = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \end{bmatrix} \qquad \boxed{\mathbf{x}_p = \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \vdots \end{bmatrix}}$$

Initialization

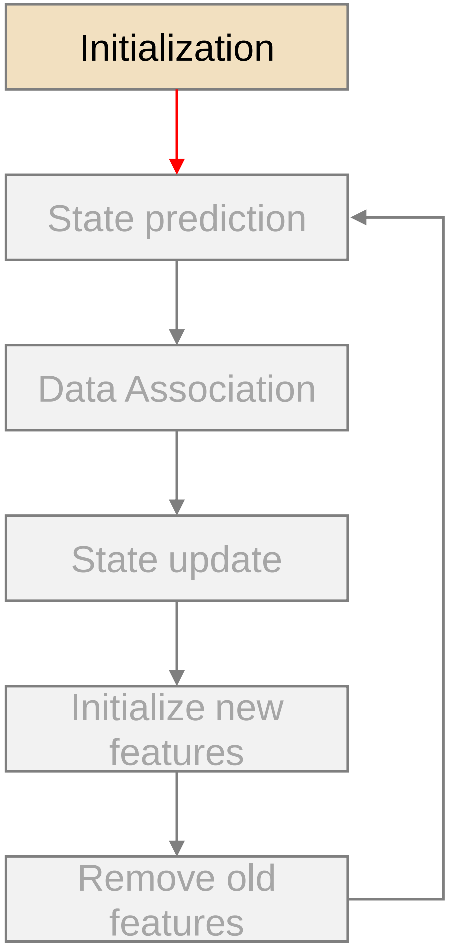


Step 1:

- Use LSD line detector* to detect line segments on the image.

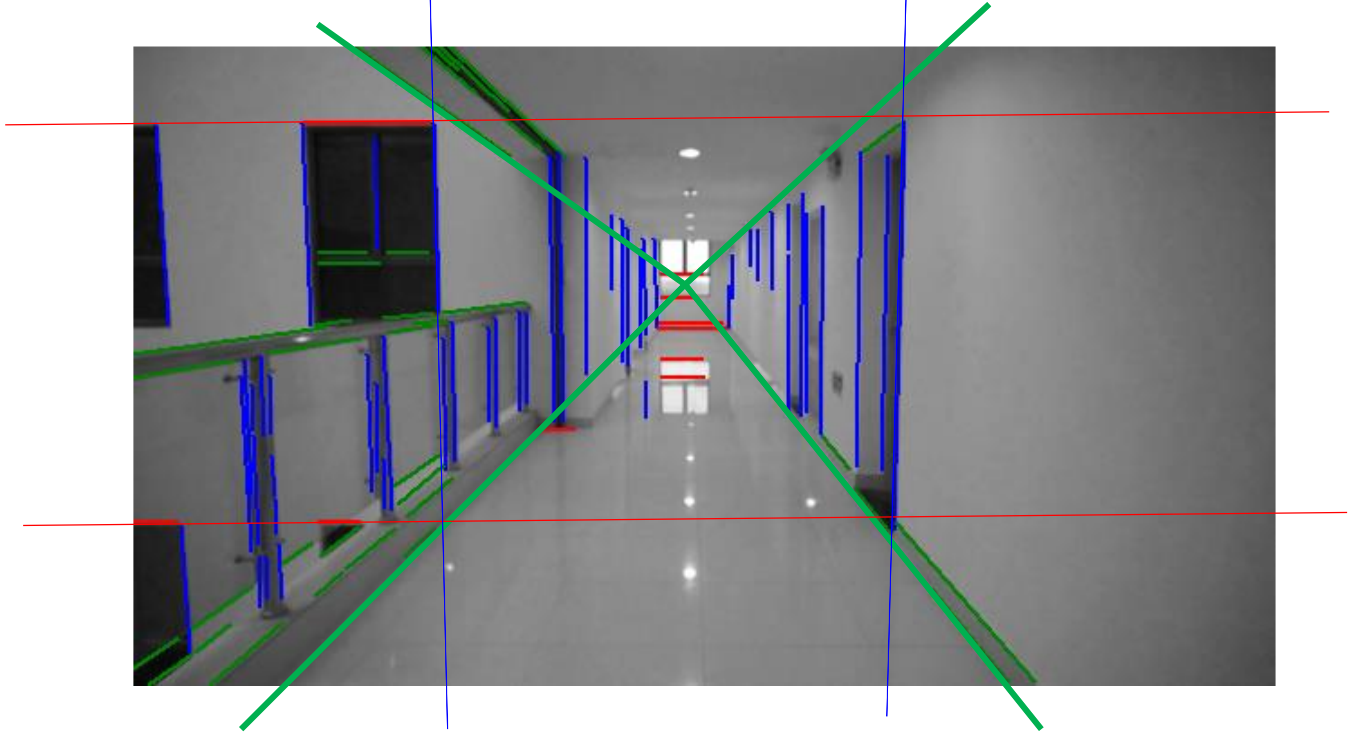
* Von Gioi, Rafael Grompone, et al. "LSD: A fast line segment detector with a false detection control." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 4 (2008): 722-732.

Initialization



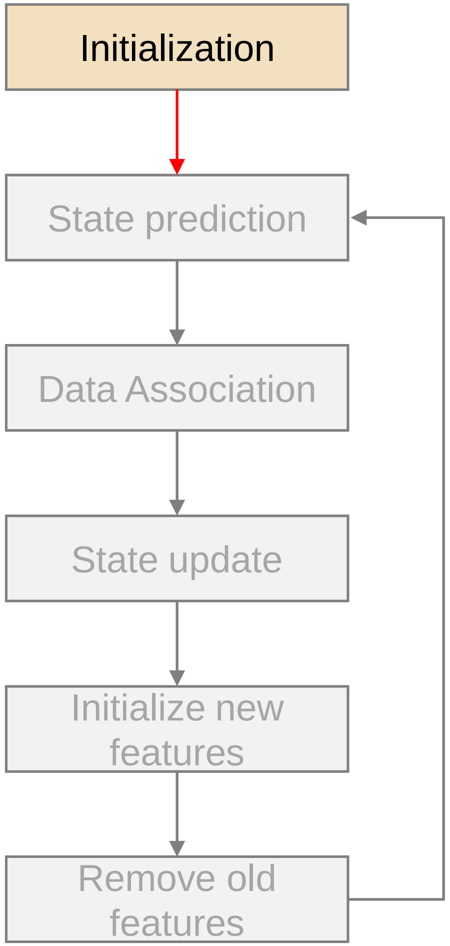
Step 2:

- Apply J-linkage* to classify parallel line segments into groups and detect vanishing points



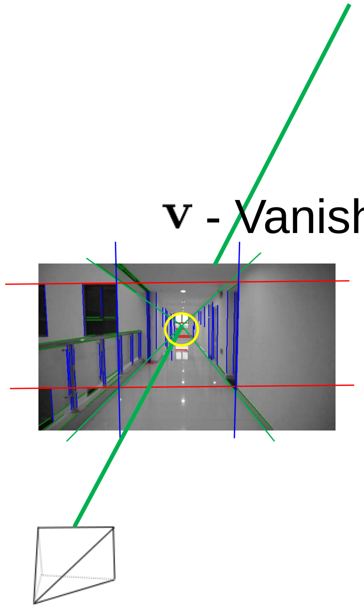
*Toldo, Roberto, and Andrea Fusiello. "Robust multiple structures estimation with j-linkage." *Computer Vision–ECCV 2008*. Springer Berlin Heidelberg, 2008. 537-547.

Initialization



Step 3:

- Estimate the dominant direction from the vanishing points.



$$\mathbf{v} = \mathbf{K}\mathbf{R}^{cw}\eta$$

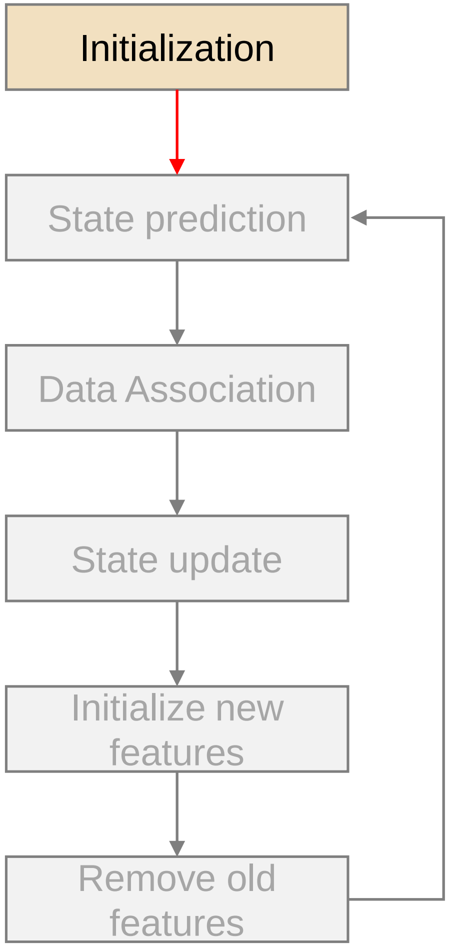
$$\eta \propto \mathbf{R}^{wc}\mathbf{K}^{-1}\mathbf{v}$$

$\mathbf{K}$: Camera intrinsic

$\mathbf{R}^{cw}$: Rotation from the world frame to the camera frame

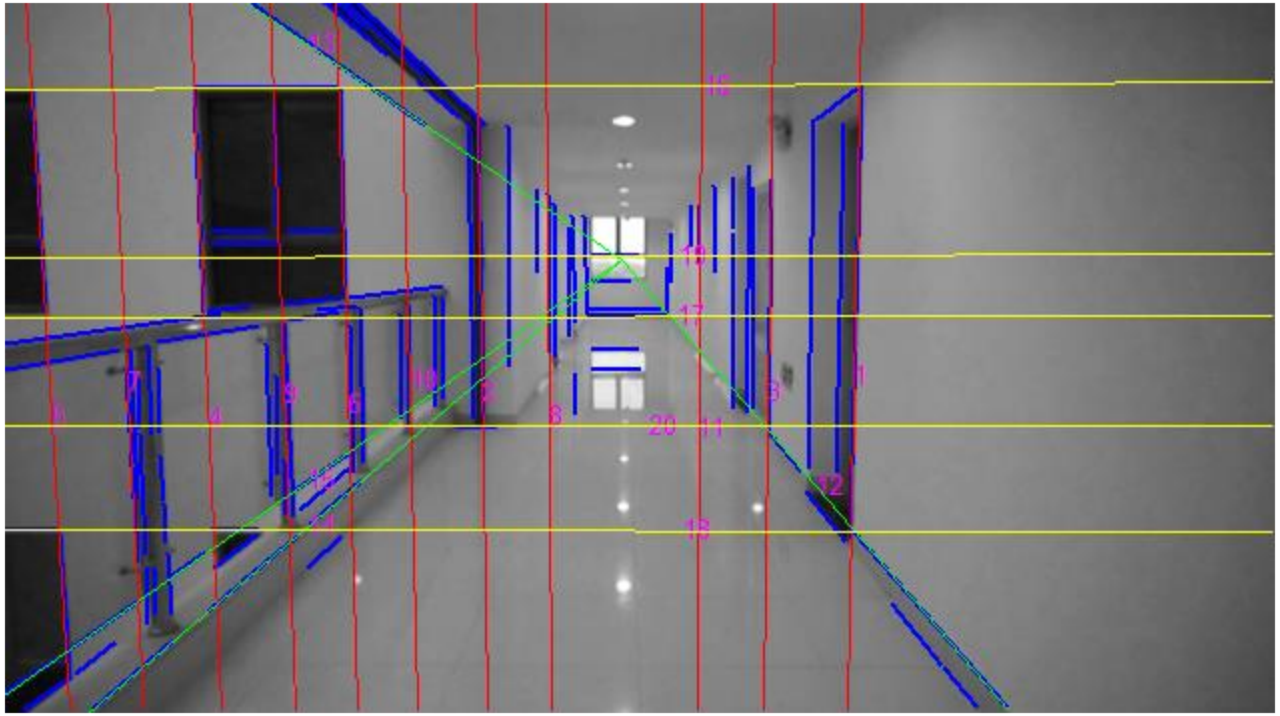
$$\mathbf{R}^{wc} = (\mathbf{R}^{cw})^T$$

Initialization

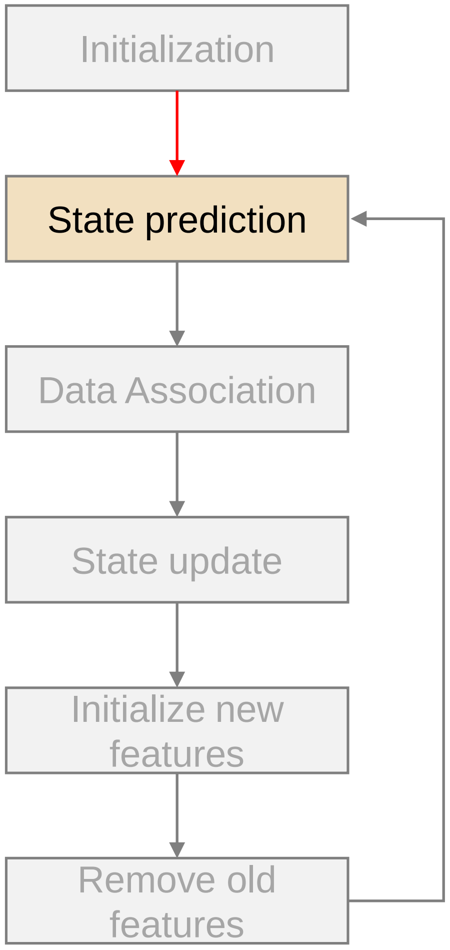


Step 4:

- Refine the dominant directions by non-linear least square optimization
- Initialize new lines (See the feature management section)



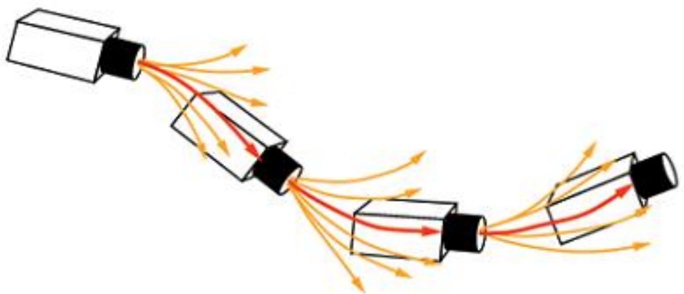
State prediction



- Camera – constant velocity or odometer data (if available)

$$\mathbf{f}_c(\mathbf{x}_c) = \begin{pmatrix} \bar{\mathbf{p}}^w \\ \bar{\mathbf{q}}^{wc} \\ \bar{\mathbf{v}}^w \\ \bar{\omega}^c \end{pmatrix} = \begin{pmatrix} \mathbf{p}^w + \mathbf{v}^w \Delta t \\ \mathbf{q}^{wc} \cdot \mathbf{q}(\omega^c) \Delta t \\ \mathbf{v}^w \\ \omega^c \end{pmatrix} .$$

Odometer velocity

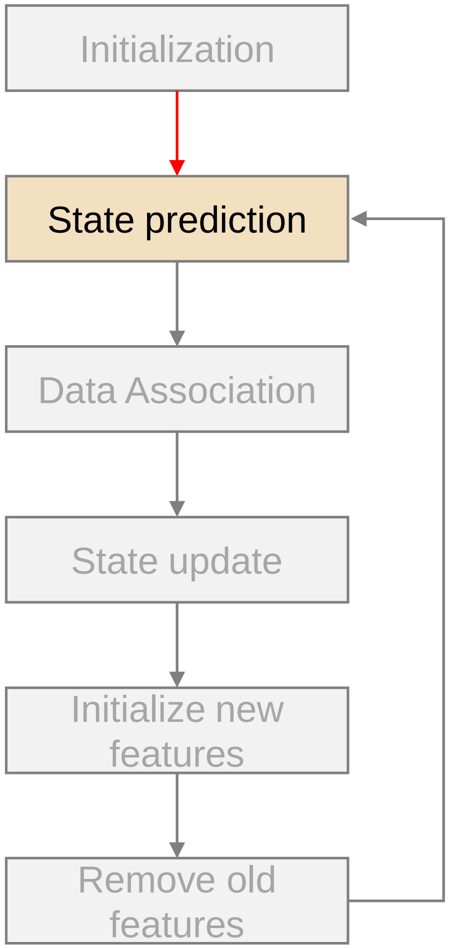


- Landmarks – static

$$\mathbf{f}_p(\mathbf{x}_p) = \mathbf{x}_p \qquad \mathbf{f}_l(\mathbf{x}_l) = \mathbf{x}_l$$

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{f}_c(\mathbf{x}_c) \\ \mathbf{x}_p \\ \mathbf{x}_l \end{bmatrix} + \mathbf{n}$$

State prediction



Covariance propagation

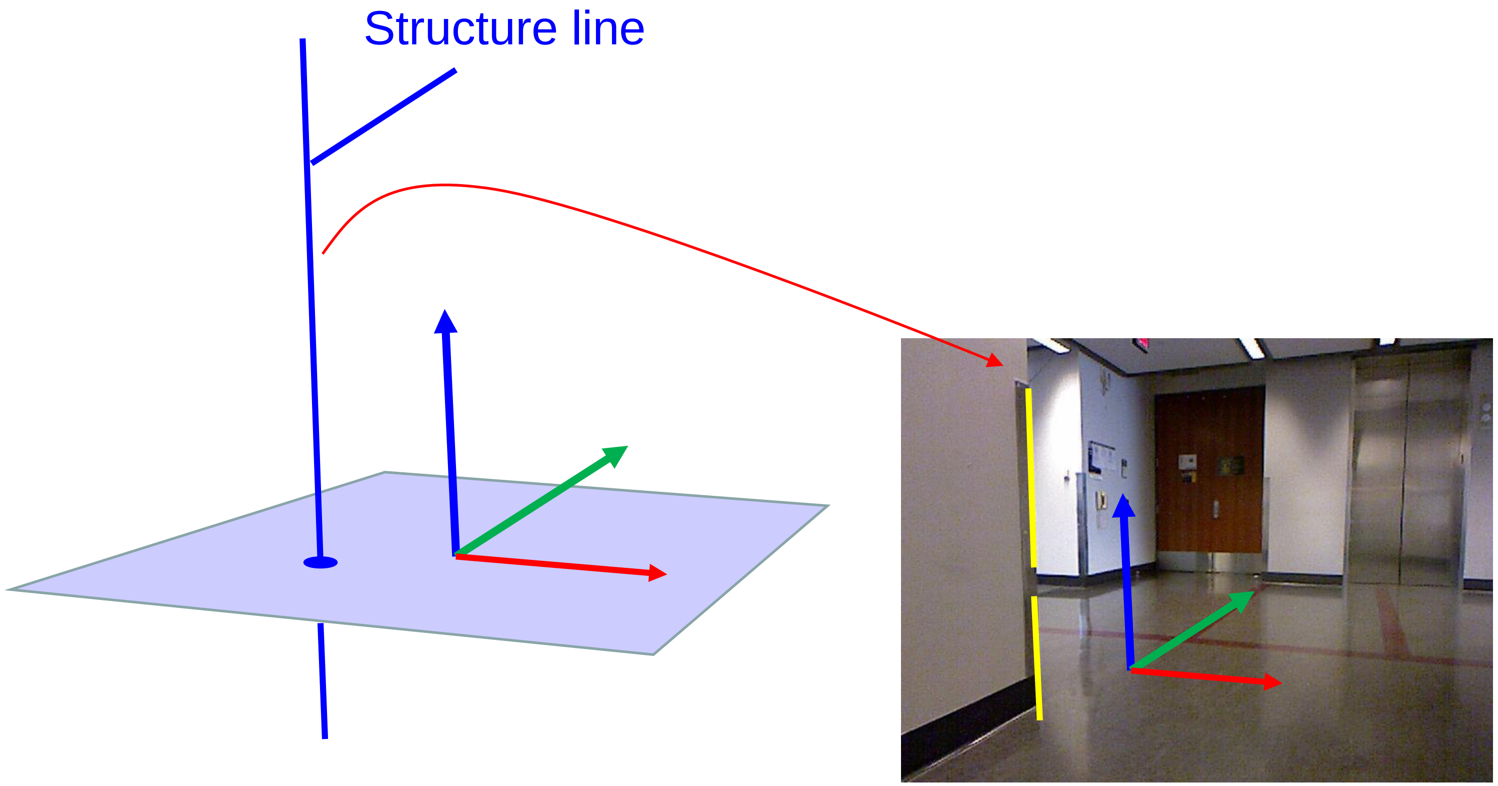
$$\bar{\boldsymbol{\Sigma}} = \mathbf{F_x} \boldsymbol{\Sigma} \mathbf{F_x}^T + \mathbf{F_n} \boldsymbol{\Sigma_n} \mathbf{F_n}^T$$

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{f}_c(\mathbf{x}_c) \\ \mathbf{x}_p \\ \mathbf{x}_l \end{bmatrix} + \mathbf{n}$$

$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial \mathbf{f}_c}{\partial \mathbf{x}_c} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \qquad \mathbf{F_n} = \begin{bmatrix} \frac{\partial \mathbf{f}_c}{\partial \mathbf{n}} \\ \mathbf{0} \end{bmatrix}$$

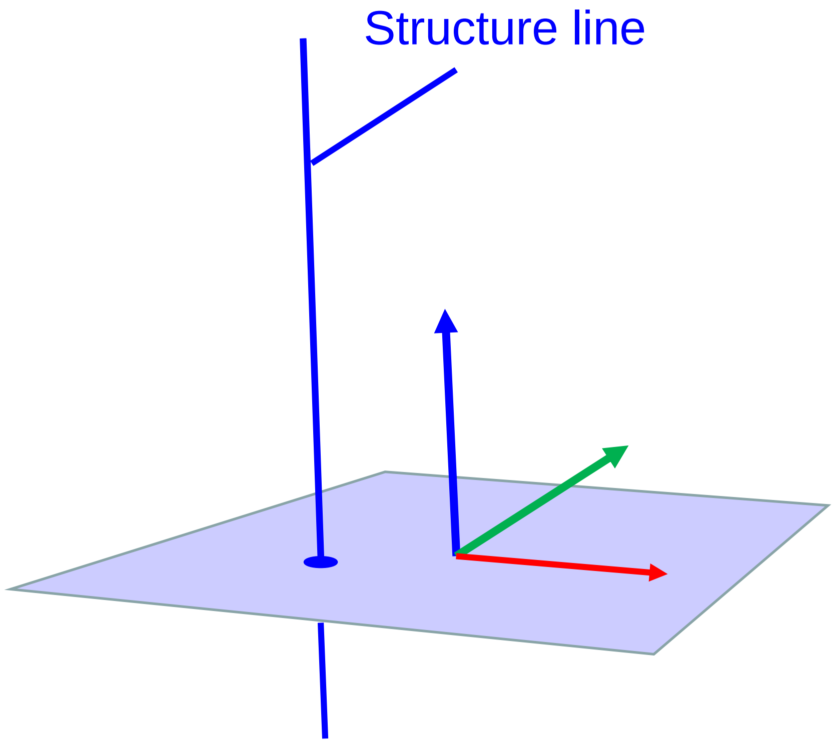
Data association

- Find the line segments corresponding to the structure line

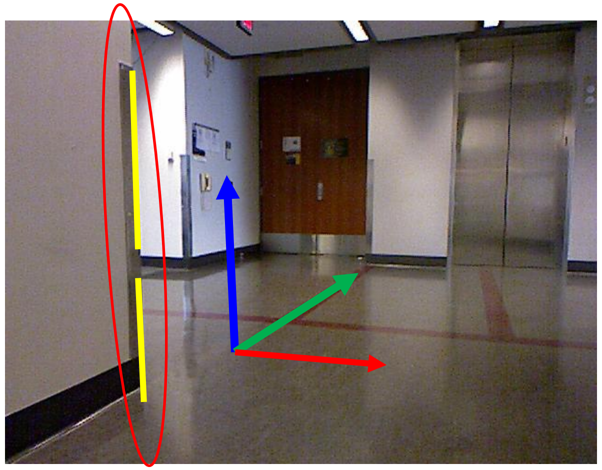


Data association

- Find the line segments corresponding to the structure line

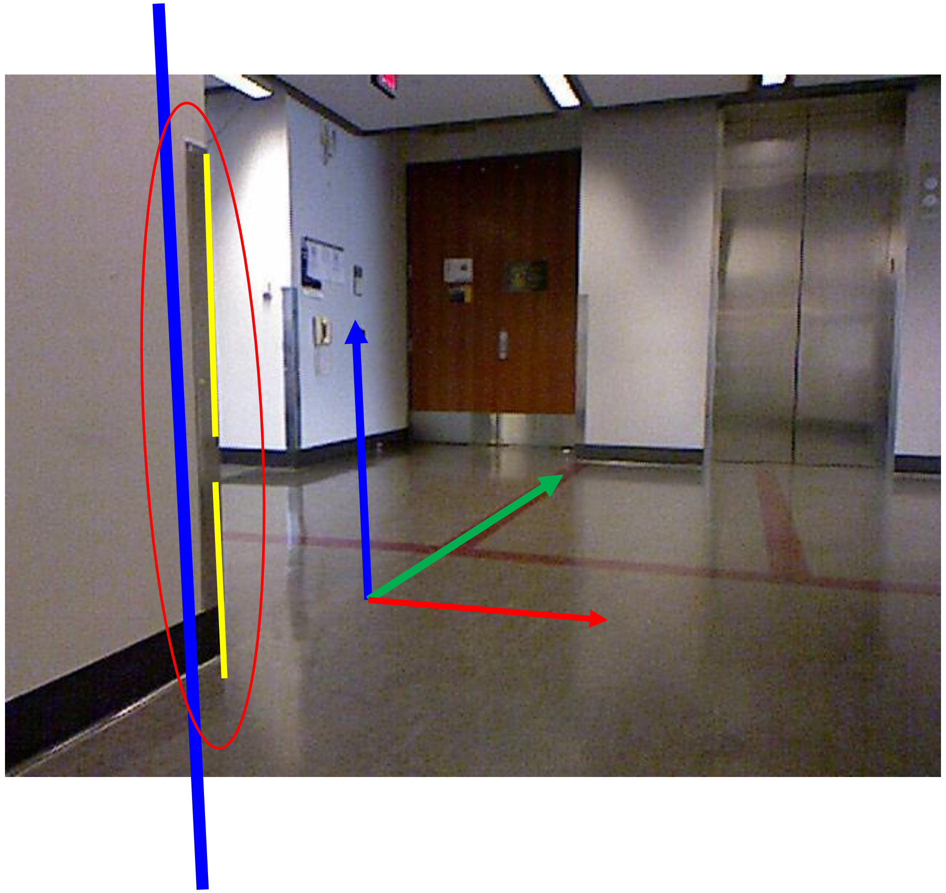


- One-to-multiple matching
- There could be false matchings (outliers).



Data association

- Step1 : Get candidate matching by χ^2-distance



$$\chi^2 = \mathbf{r}_i^T (\mathbf{H}_i \mathbf{H}_i^T)^{-1} \mathbf{r}_i^T$$

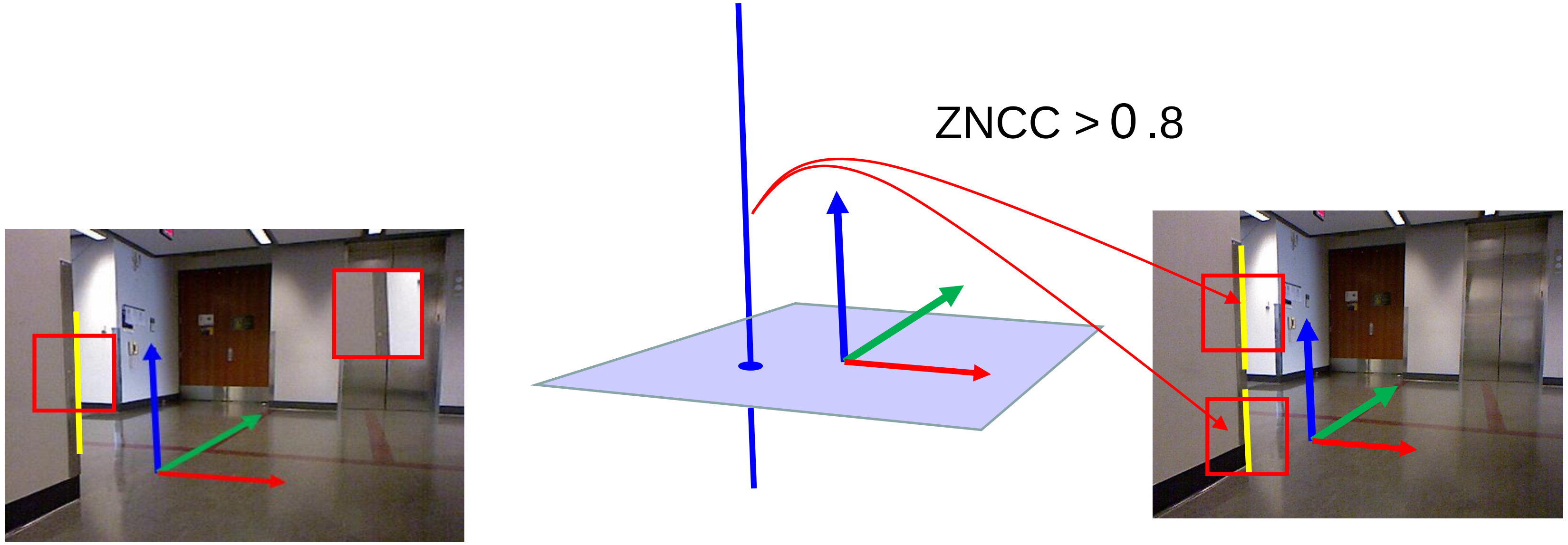
$\mathbf{r}_i$: residual vector

$\mathbf{H}_i$:Jacobian of observation function

$$\chi^2 < 5.99 \quad (Probablity > 95\%)$$

Data association

- Step 2: Comparing appearance by ZNCC (zero mean normalized cross-correlation)

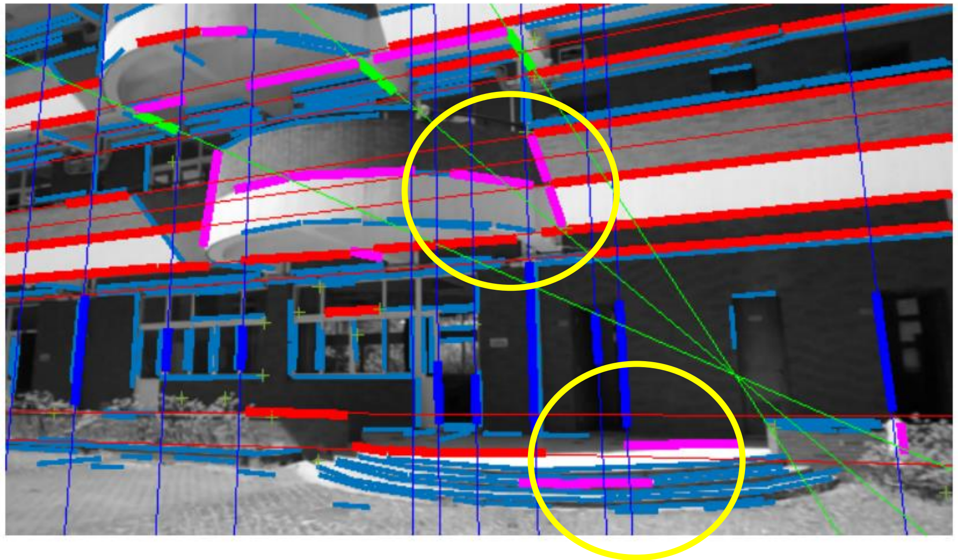


Previous frame

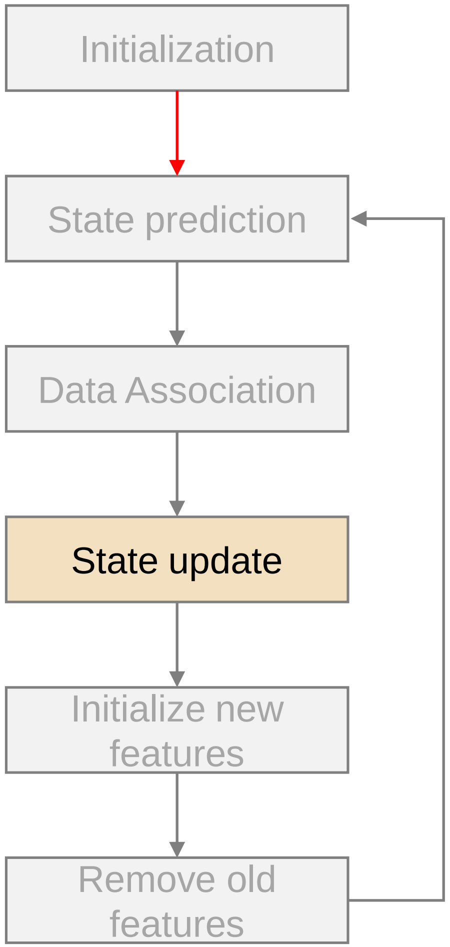
Current frame

Data association

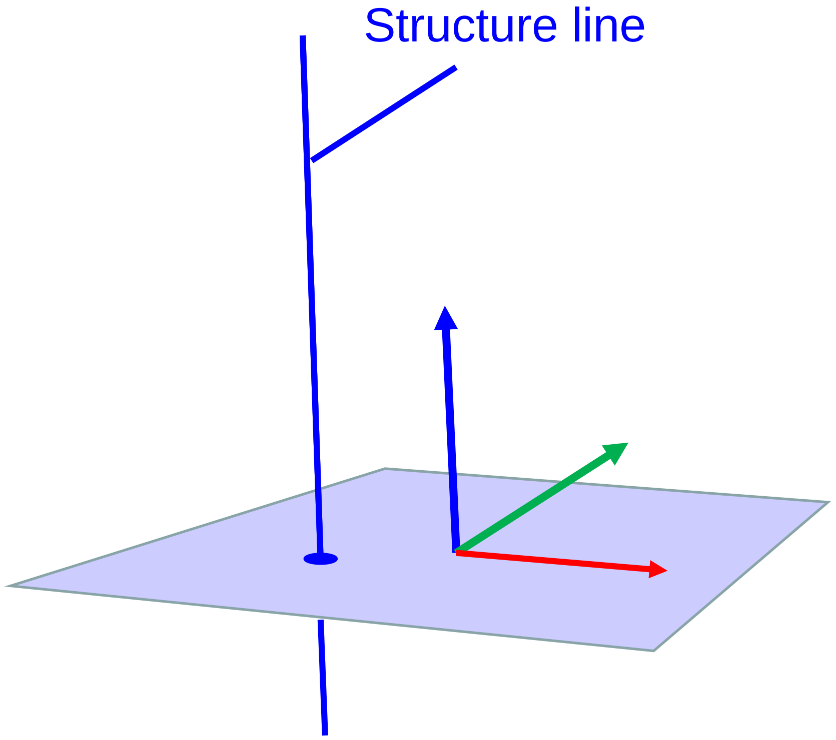
- Step3 : One-feature RANSAC to eliminate false matchings (outliers):
  - Randomly sample a candidate matching
  - Run a tentative EKF update using the sampled matching and check the number of inliers
  - Keep the inlier set with the maximum nu
  - Repeat the above steps
  - Use the inlier set to run EKF update.

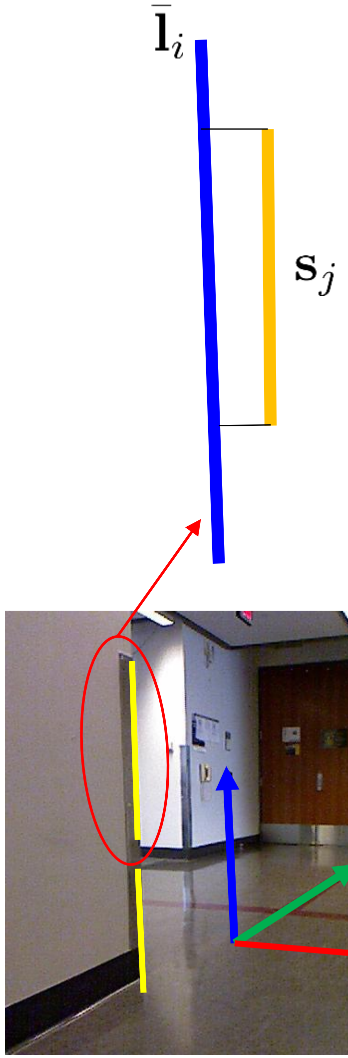


State update



Observation model:

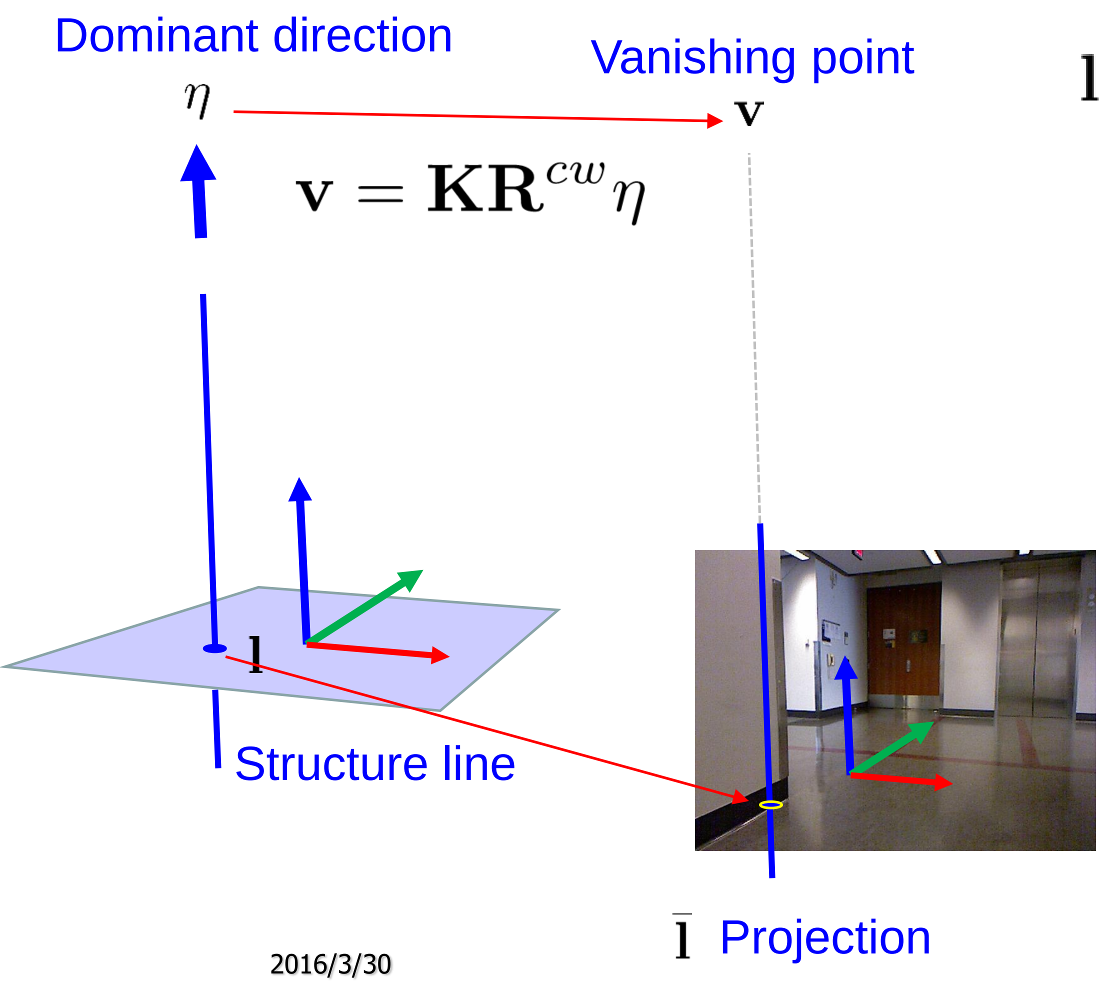




$$\mathbf{m}_{ij} = \begin{bmatrix} \mathbf{s}_j^a \cdot \bar{\mathbf{l}}_i / \sqrt{(\bar{l}_i^1)^2 + (\bar{l}_i^2)^2} \\ \mathbf{s}_j^b \cdot \bar{\mathbf{l}}_i / \sqrt{(\bar{l}_i^1)^2 + (\bar{l}_i^2)^2} \end{bmatrix}$$

State update

- Project a structure line onto the image



$$\mathbf{l} = [c_a, c_b, \theta, h]^T$$

World frame

$$\mathbf{l}^w h = \mathbf{P}^T \left([\ c_a, c_b\ ]^T h + [\ \cos(\theta), \sin(\theta)\ ]^T \right)$$

Camera frame

$$\mathbf{l}^c = \mathbf{R}^{cw}\mathbf{l}^w h - \mathbf{R}^{cw}\mathbf{p}^w h$$

Image

$$\mathbf{l}^i = \mathbf{K}\mathbf{l}^c$$

$$\bar{\mathbf{l}} = \mathbf{v} \times \mathbf{l}^i$$

Connect with vanishing point

State update

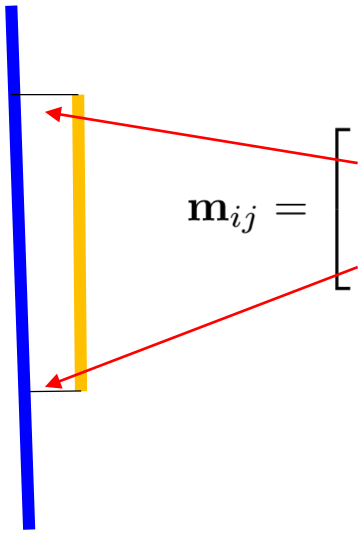
Observation model

- Observation function (measurement function):

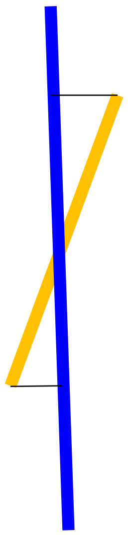
$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \vdots \\ \mathbf{m}_{ij} \\ \vdots \end{bmatrix}$$

- Since the desired distance is zero, the residual is computed as :

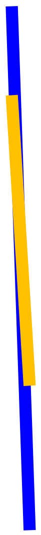
$$\mathbf{r}(\mathbf{x}) = -\mathbf{h}(\mathbf{x})$$



$$\mathbf{m}_{ij} = \begin{bmatrix} \mathbf{s}_j^a \cdot \bar{\mathbf{l}}_i / \sqrt{(\bar{l}_i^1)^2 + (\bar{l}_i^2)^2} \\ \mathbf{s}_j^b \cdot \bar{\mathbf{l}}_i / \sqrt{(\bar{l}_i^1)^2 + (\bar{l}_i^2)^2} \end{bmatrix}$$



Large residual



Small residual

State update

Standard Extended Kalman Filter:

Predicted state and covariance: $\bar{\mathbf{x}}, \bar{\boldsymbol{\Sigma}}$

$$\downarrow$$

$$\begin{aligned} \text{Innovation covariance:} \quad & \mathbf{S} = \mathbf{H}\bar{\boldsymbol{\Sigma}}\mathbf{H}^T + \mathbf{N} \\ \text{Kalman gain:} \quad & \mathbf{K} = \bar{\boldsymbol{\Sigma}}\mathbf{H}^T\mathbf{S}^{-1} \end{aligned}$$

$$\downarrow$$

$$\begin{aligned} \text{State update:} \quad & \mathbf{x} \leftarrow \bar{\mathbf{x}} + \mathbf{K}\mathbf{r} \\ \text{Covariance update:} \quad & \boldsymbol{\Sigma} \leftarrow \bar{\boldsymbol{\Sigma}} - \mathbf{K}\mathbf{S}\mathbf{K}^T \end{aligned}$$

$$\downarrow$$

$\mathbf{x}, \boldsymbol{\Sigma}$

$\mathbf{H} = \frac{\partial \mathbf{m}}{\partial \mathbf{x}}$:Jacobian of observation function

N : Observation noise - Uncertainty of detected line segments

$$\begin{bmatrix} \ddots & & & \\ & 4 & & \\ & & 4 & \\ & & & \ddots \end{bmatrix}$$

State update

Initialization

State prediction

Data Association

State update

Initialize new features

Remove old features

- Multiple-pass EKF update (Point + Structure lines)

$\bar{\mathbf{x}}, \bar{\Sigma}$

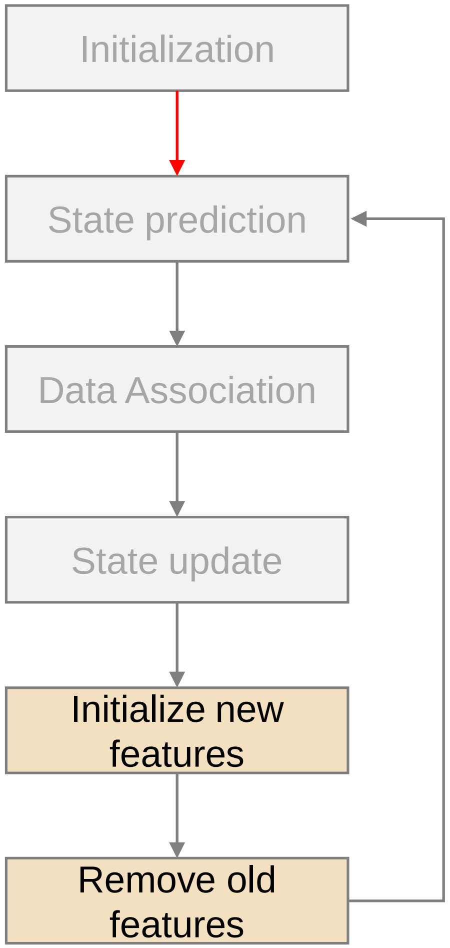
State update using point observation

$\bar{\mathbf{x}}', \bar{\Sigma}'$

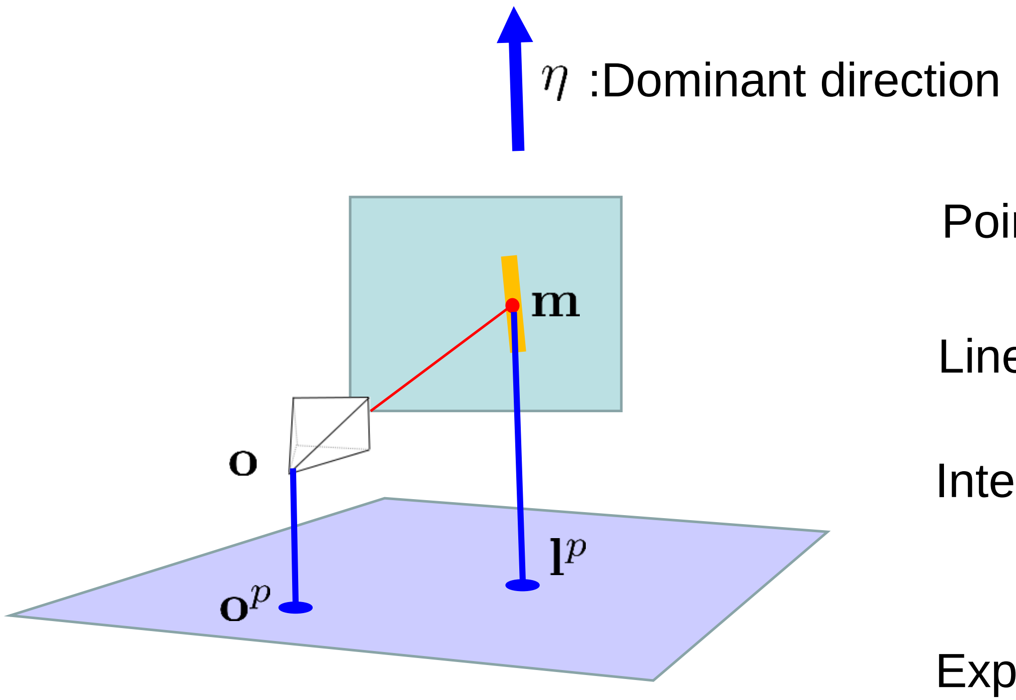
State update using structure line observation

$\mathbf{x}, \Sigma$

Feature management



Initialize new structure lines



Point in world frame: $\mathbf{m} = \mathbf{R}^{wc}\mathbf{K}^{-1}\tilde{\mathbf{m}} + \mathbf{p}^w$

Line though the point: $\mathbf{L} = \mathbf{m}\eta^T - \eta\mathbf{m}^T$

Intersection with parameter plane:

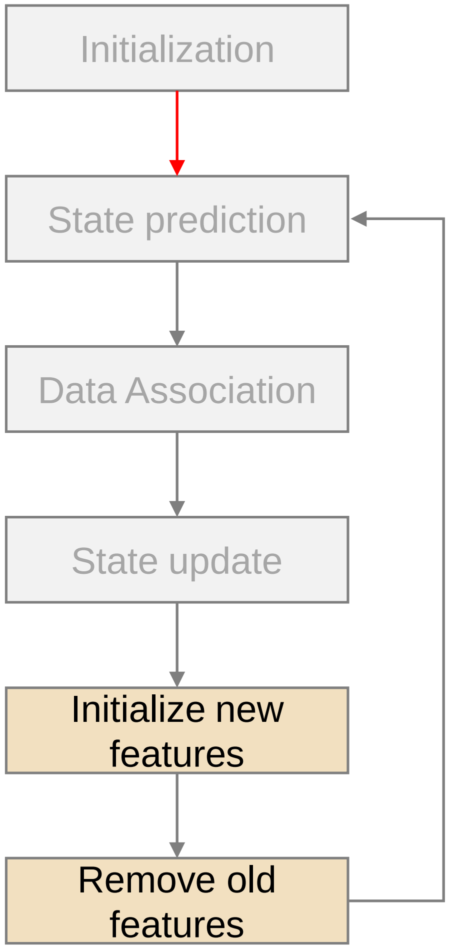
$$\tilde{\mathbf{l}}^w = \mathbf{L}\pi$$

Expressed in parameter plane (xy-plane):

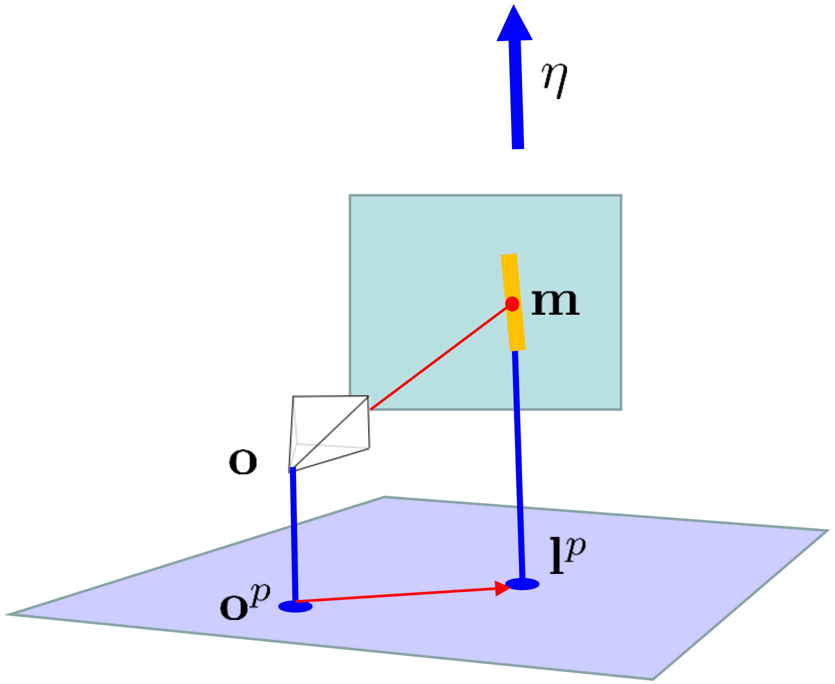
$$\mathbf{l}^p = \mathbf{P}\mathbf{l}^w$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

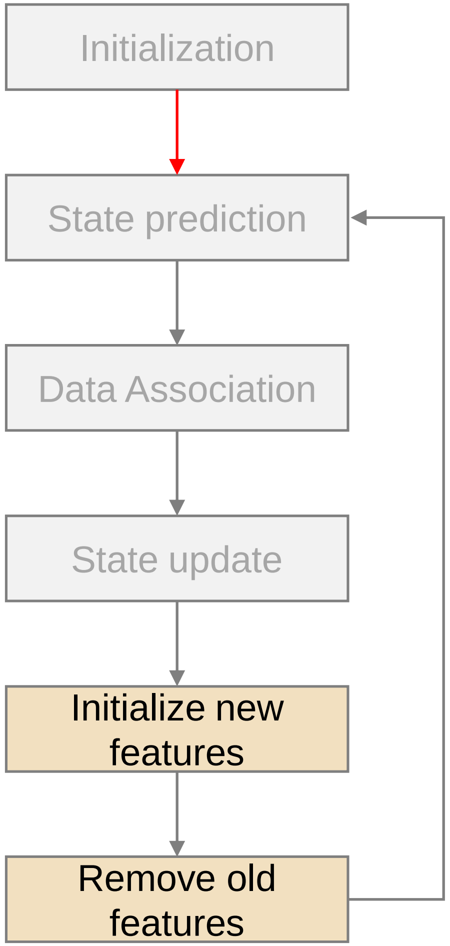
Feature management



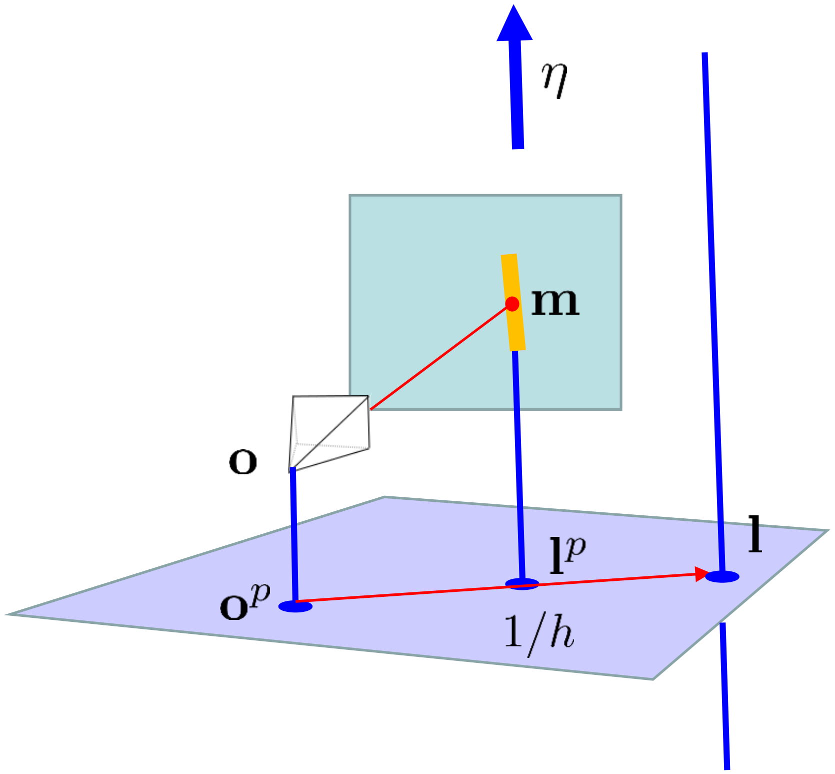
Initialize new structure lines



Feature management

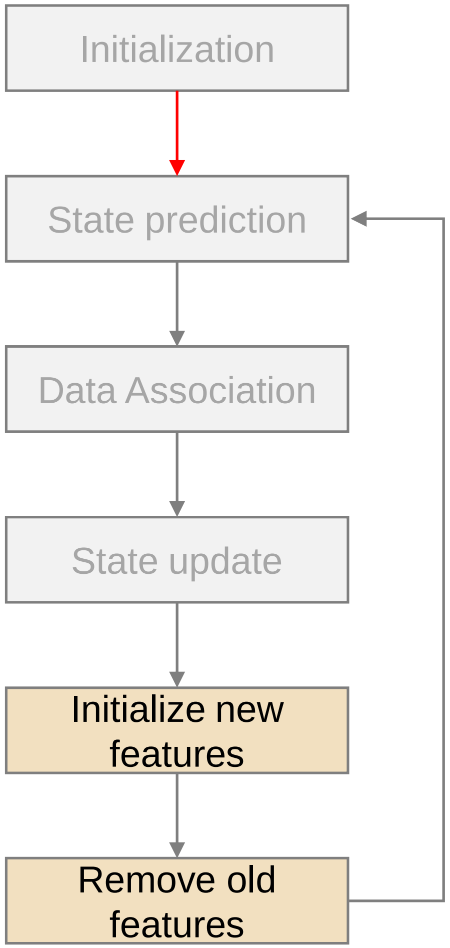


Initialize new structure lines



$$\mathbf{l} = [\ c_a, c_b, \theta, h\ ]^T$$
$$= \left[\ \mathbf{o}^p(1), \mathbf{o}^p(2), atan\left(\frac{\mathbf{l}^p(2)-\mathbf{o}^p(2)}{\mathbf{l}^p(1)-\mathbf{o}^p(1)}\right), h_0\ \right]^T$$

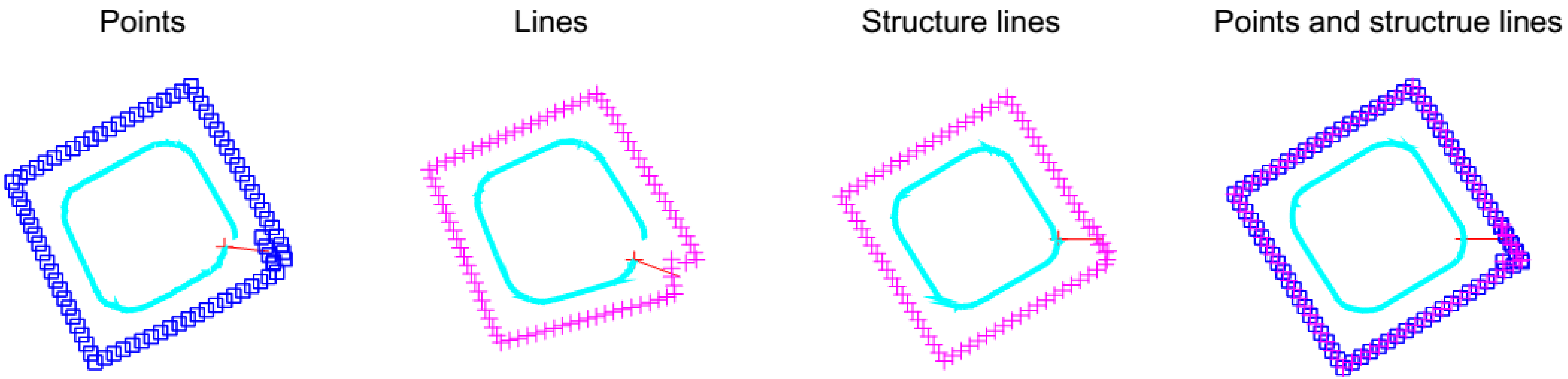
Feature management



- The number of features is limited in the state.
- For each dominant direction, we keep a maximum number of structure lines.
- Old features are removed according to the number of matching failure (NOF)

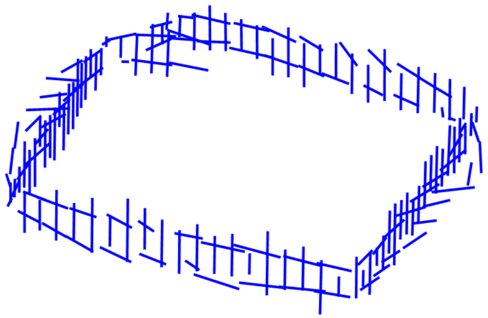
Results

Simulated case

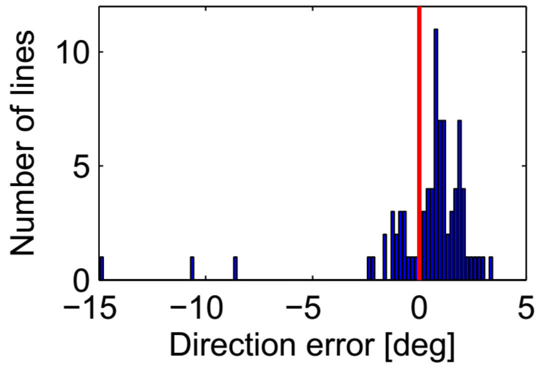


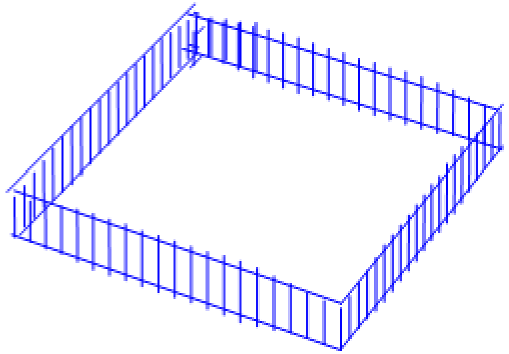
Results

Simulated case



Lines





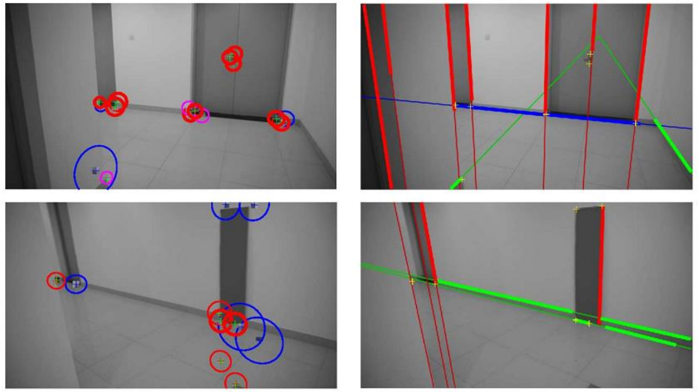
Structure lines

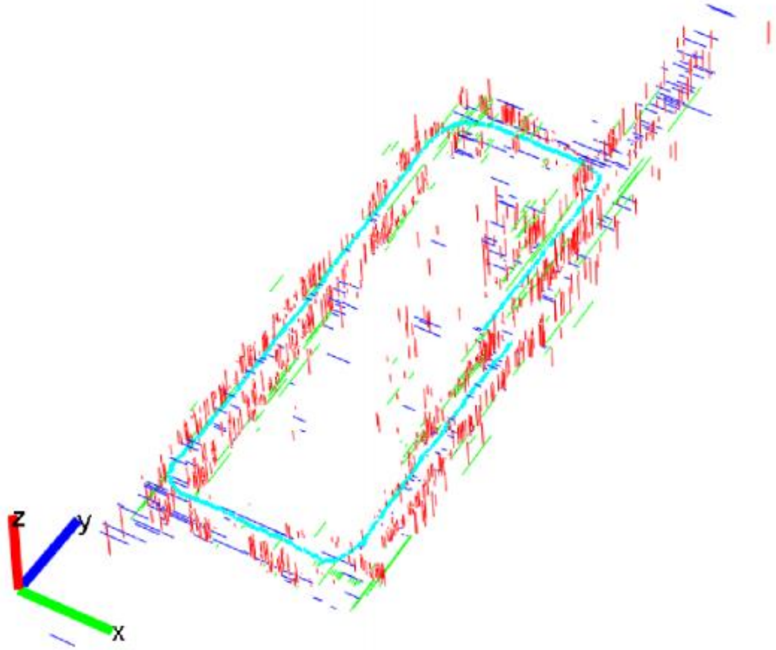
Lines V.S. structure lines

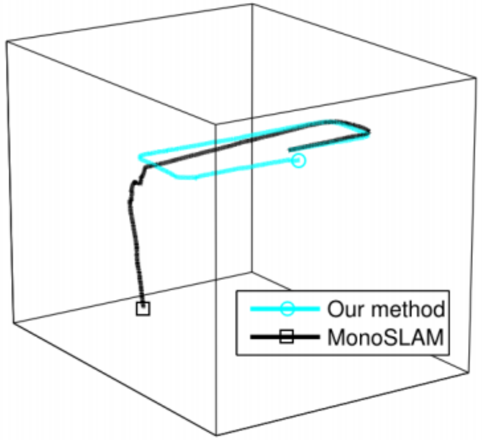
Results

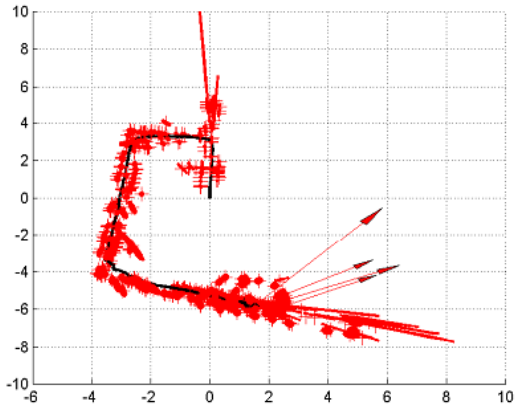
Real-world case (Using one video camera)

- Indoor texture-less scenes





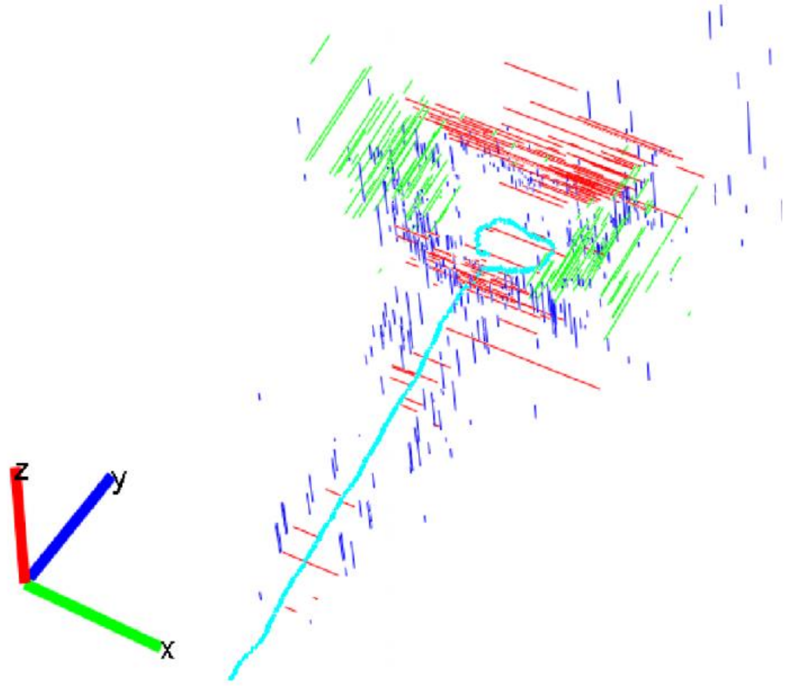


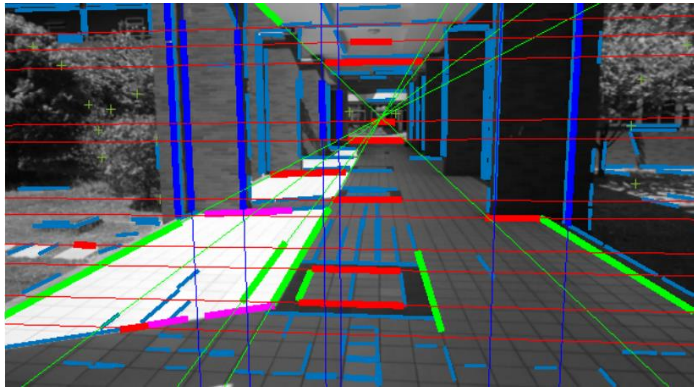


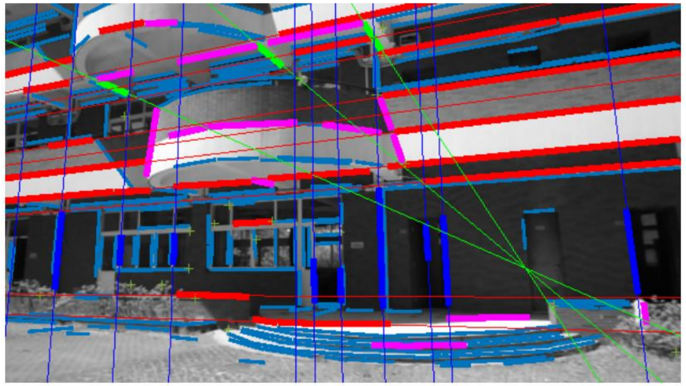
Results of real-word indoor scenes

Real world case (Using one video camera)

- Outdoor texture rich scenes



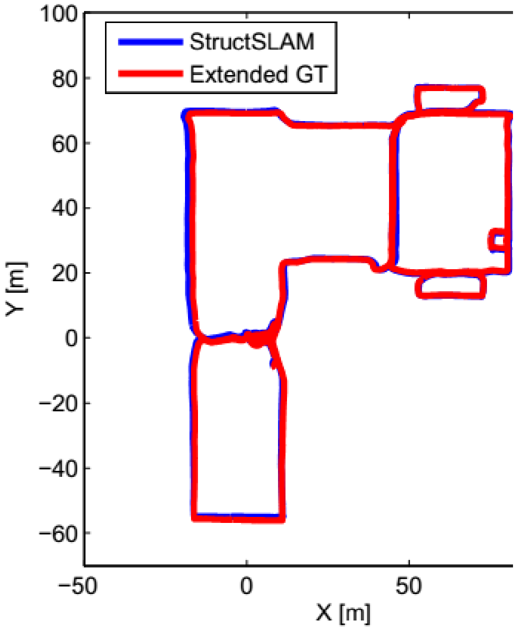




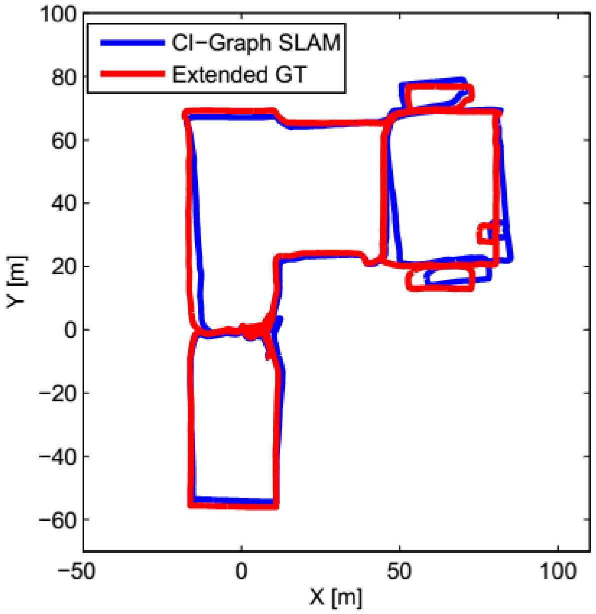
Benchmark datasets

Rawseeds datasets (all methods fused the odometer information)

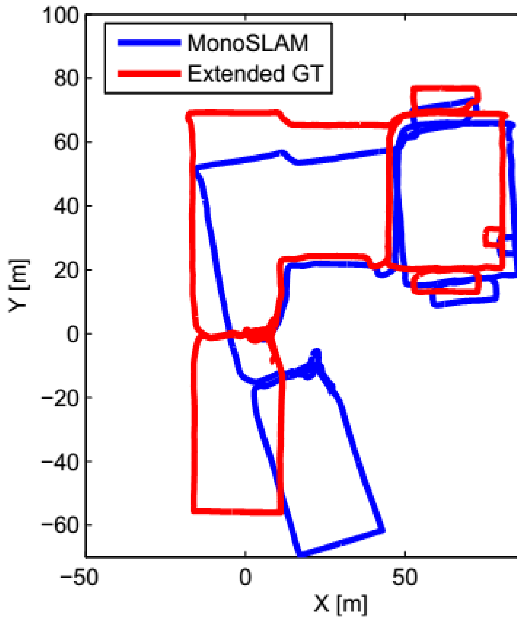
- **Bicacco-02-25b** : A 774m trajectory in the indoor scene



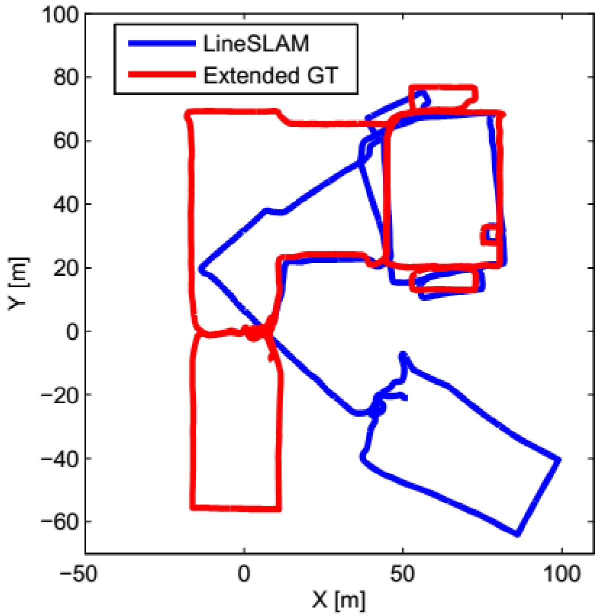
(a) StructSLAM vs GT



(b) CI-Graph vs GT

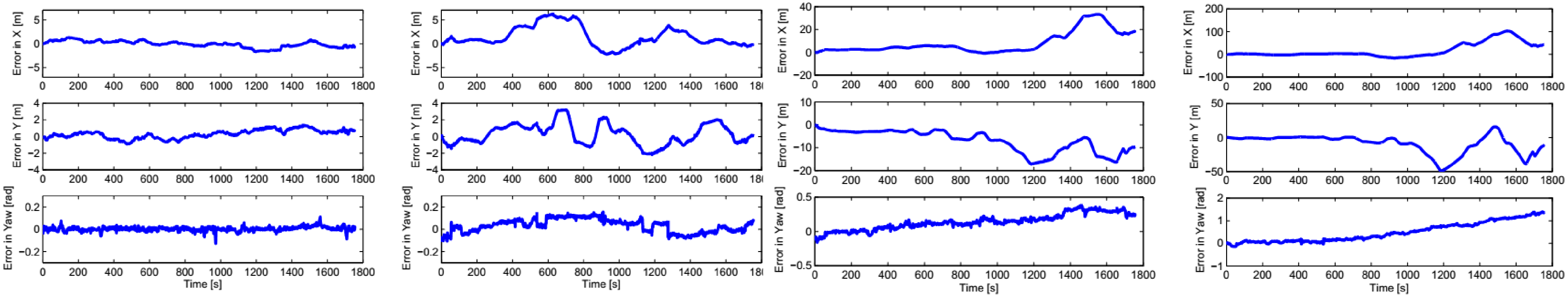


(c) MonoSLAM vs GT

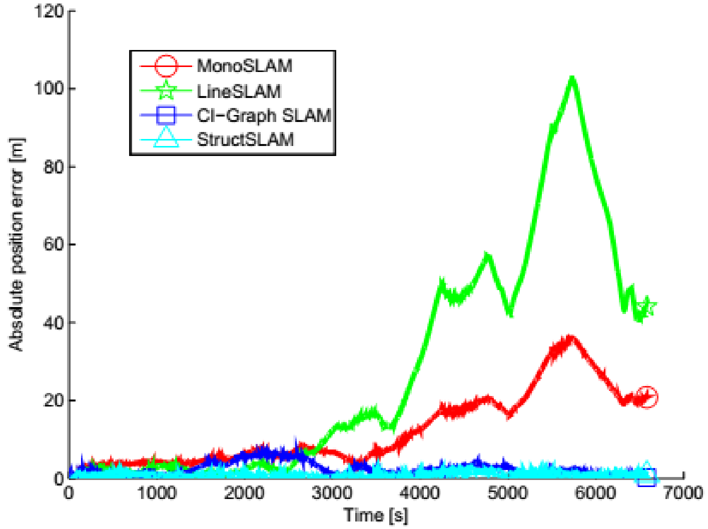


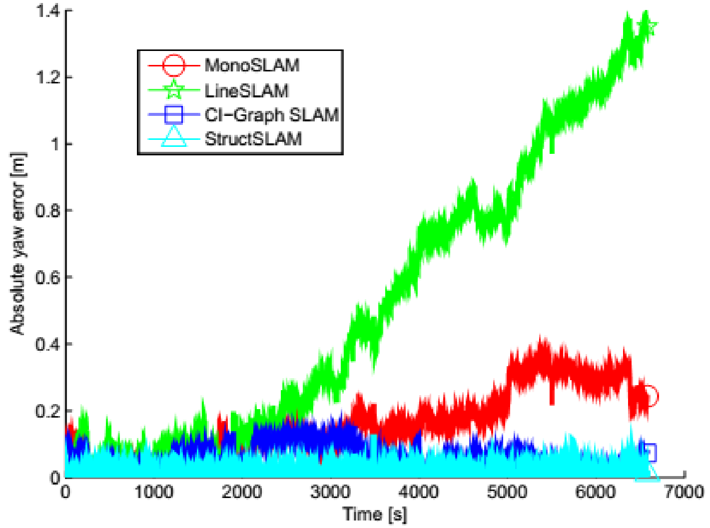
(d) LineSLAM vs GT

Benchmark results



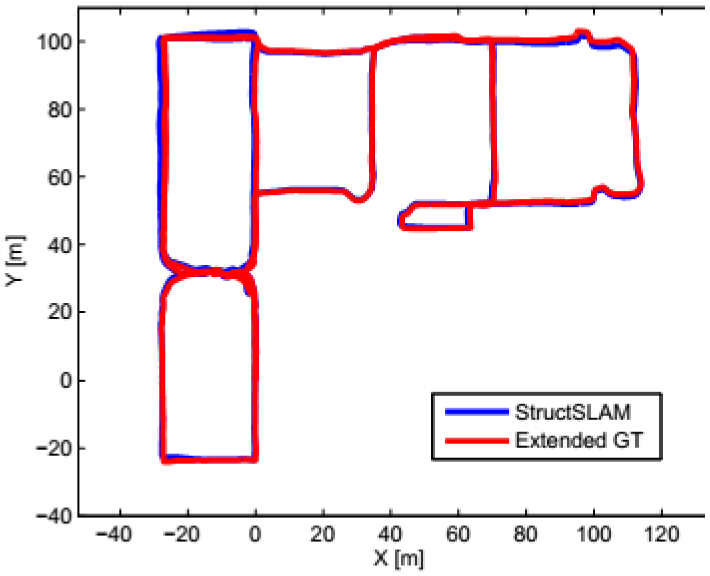
(a) StructSLAM (b) CI-Graph (c) MonoSLAM (d) LineSLAM



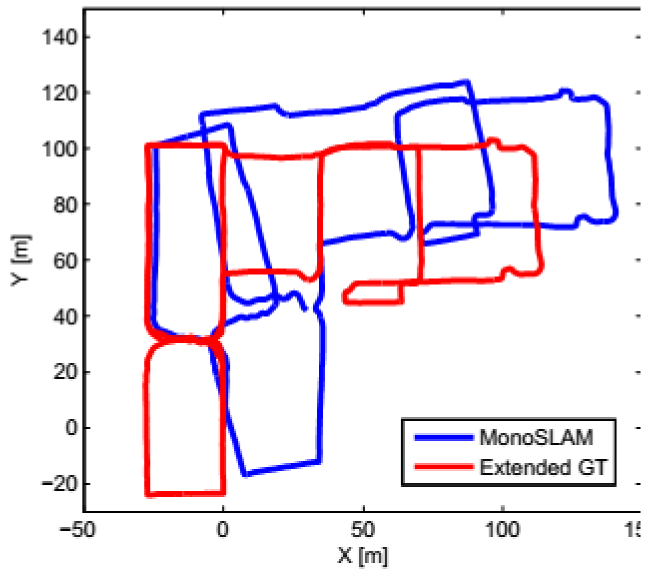


Benchmark datasets

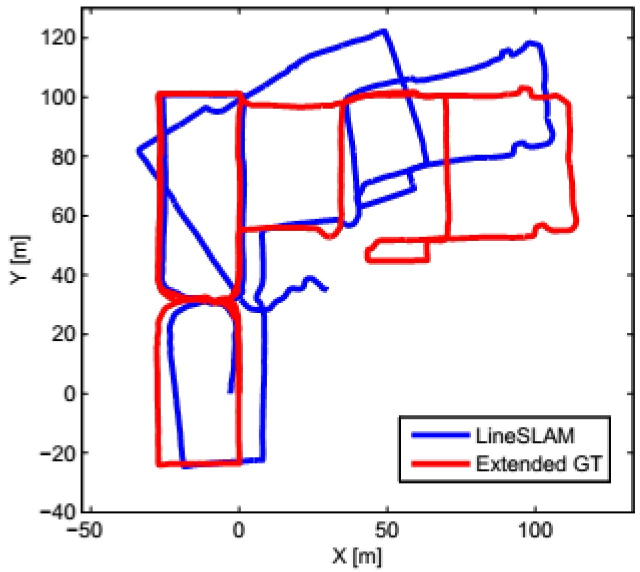
Bicacco-02-27a :



(a) StructSLAM



(b) MonoSLAM



(c) LineSLAM

Benchmark datasets

Comparision

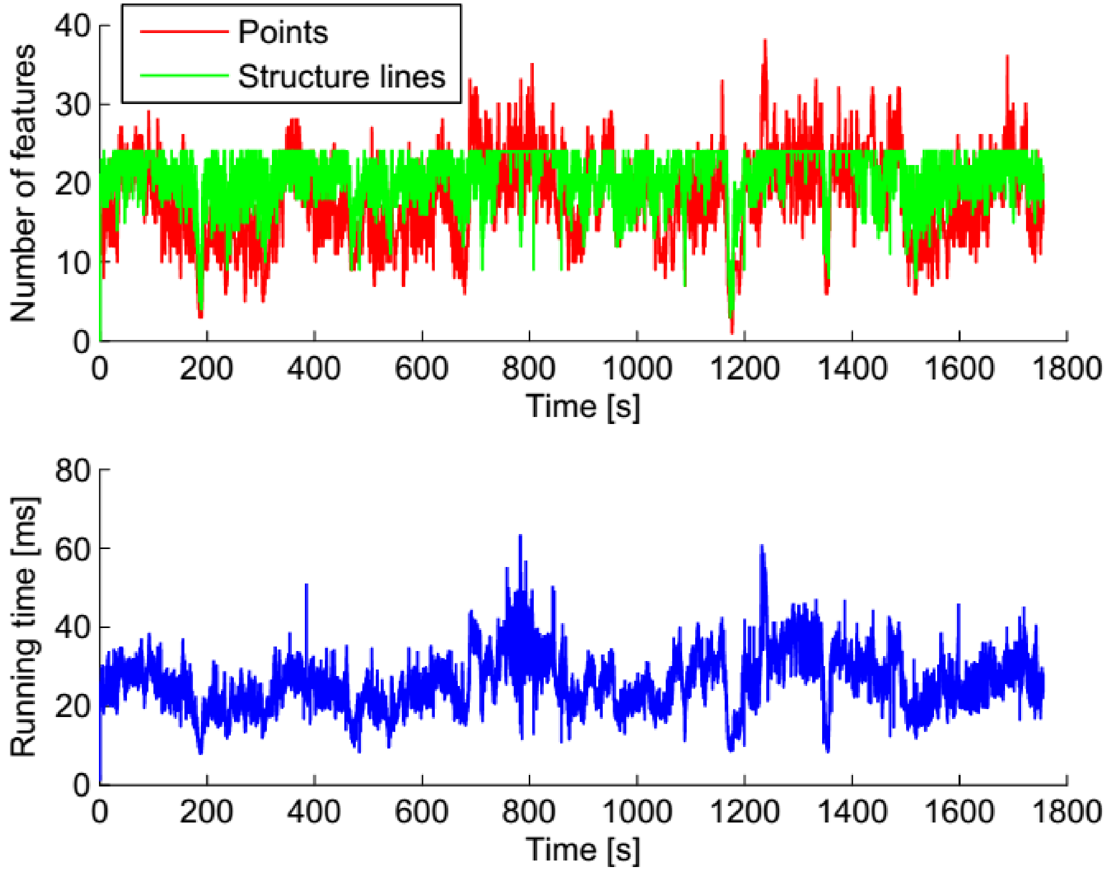
	Biccoca 25b [774 m]						Biccoca 27a [967m]					
	Position[m]			Yaw[rad]			Position[m]			Yaw[rad]		
	Mean	Max	Std	Mean	Max	Std	Mean	Max	Std	Mean	Max	Std
MonoSLAM	12.22	36.12	9.469	0.154	0.371	0.102	24.85	37.90	11.66	0.187	0.392	0.104
LineSLAM	27.63	102.7	29.65	0.509	1.393	0.420	12.77	32.35	10.17	0.208	0.706	0.206
CI-GraphSLAM	2.236	6.453	1.675	0.055	0.155	0.035	-	-	-	-	-	-
StructSLAM	**0.797**	**1.916**	**0.447**	**0.012**	**0.129**	**0.011**	**0.793**	**1.913**	**0.493**	**0.017**	**0.214**	**0.017**

TABLE II. ERROR COMPARISON.

StructSLAM : Without any loop-closing algorithms being applied!

Running time efficiency

- A common PC with i7 4-core cpu (2.7GHz) (c++ implementation)



Average running time: **25.8 ms**
Peak running time: **62.9 ms**

Conclusion & Discussion

- StructSLAM is more robust in texture-less indoor scenes than conventional SLAM methods
- With global orientation information encoded in structure lines, StructSLAM produces much less drift error.
- It is well fit for robotic and augmented reality (AR) applications in indoor scenes.

Conclusion & Discussion

- There are also some problems:
  - Line are not distinguishable as points (ZNCC does not work well for large view angle changes)
  - Dominant directions should be captured in the image at initialization stage.
  - Dominant directions is fixed after initialization

The Third UAV competition in SJTU

- AI technology (computer vision, learning, autonomous exploration) on micro drones

Coming competition (New URL) : http://drone.sjtu.edu.cn
Past competition (Old URL) : http://mediosoc.sjtu.edu.cn/wordpress

Thanks!