Parallel Tracking and Mapping for Small AR Workspaces

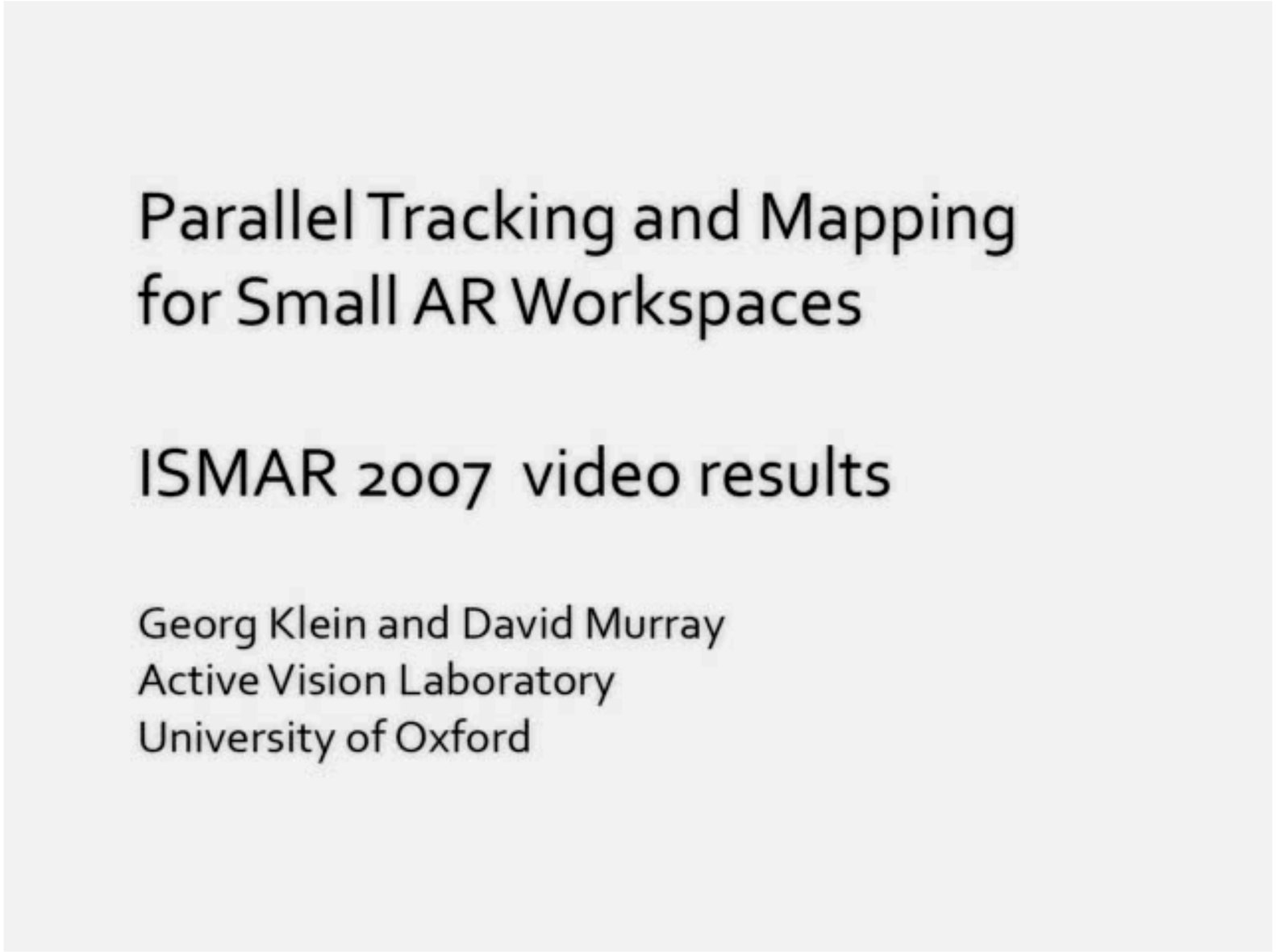
16-833: Robot Localization and Mapping

Sudharshan Suresh

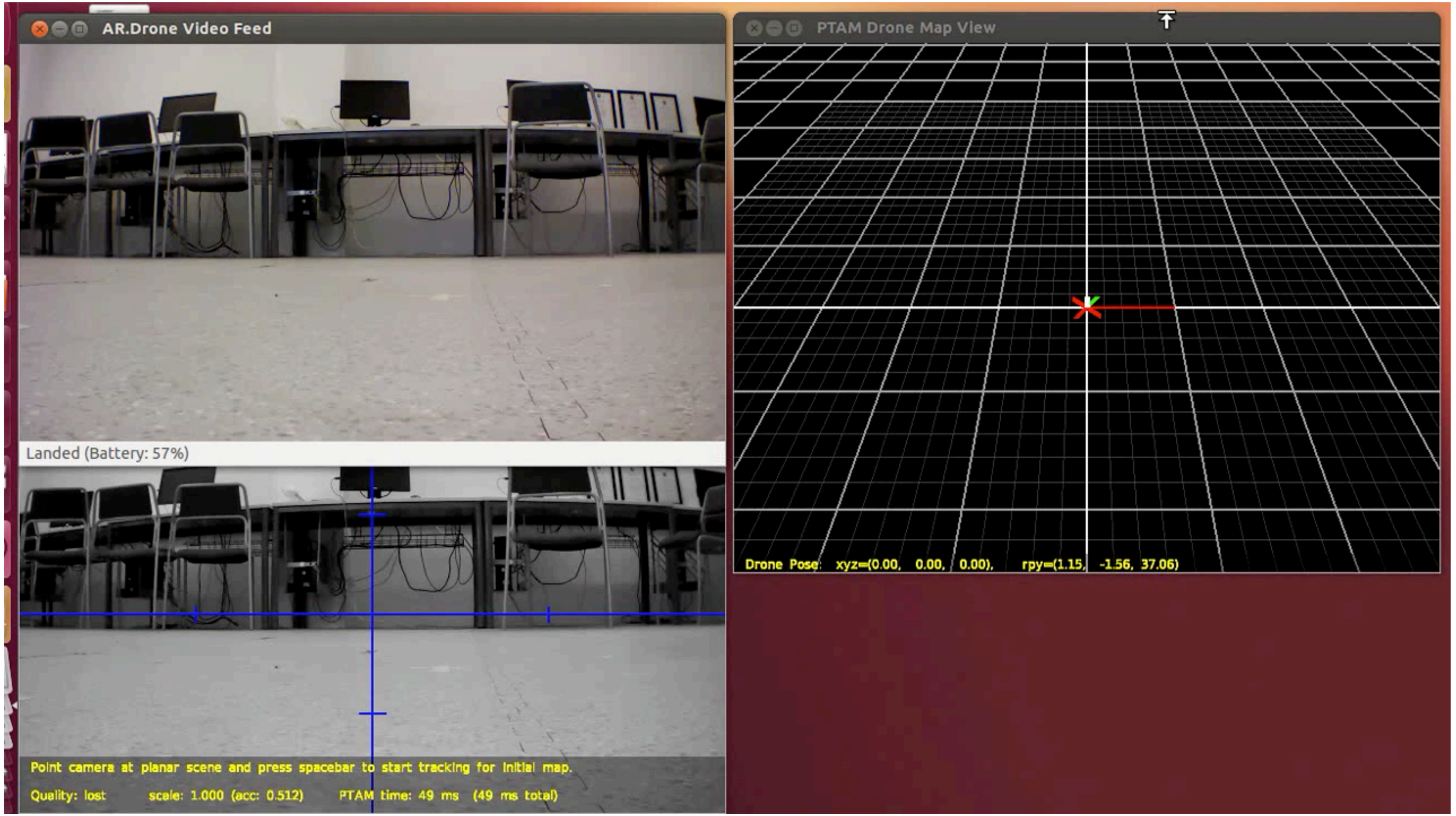
Fall 2019

Slides courtesy: Paloma Sodhi

[1] Klein, Georg, and David Murray. "**Parallel tracking and mapping for small AR workspaces.**" Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), 2007.



Parallel Tracking and Mapping for Small AR Workspaces (PTAM) - https://www.youtube.com/watch?v=F3s3M0mokNc



AR Drone fly with PTAM - https://www.youtube.com/watch?v=59TWIjDYmB8

Outline

Introduction

- Motivation
- Key components

Tracking Thread

- Frame Acquisition
- Points Projection
- Patch Search
- Pose Update

Mapping Thread

- Map Initialization
- Keyframe Insertion
- Map Optimization

Results and Conclusions

Introduction

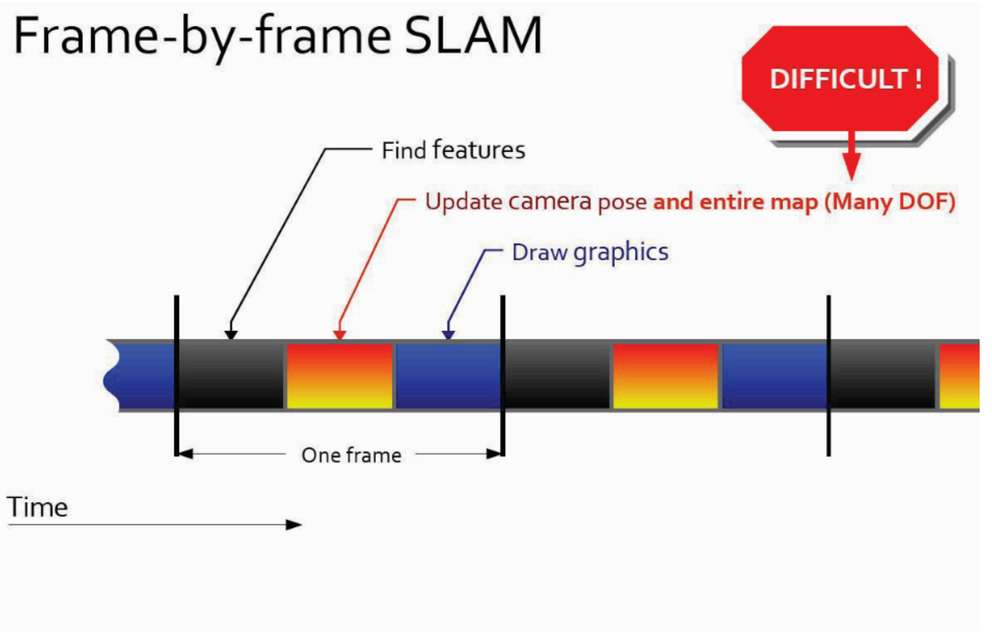
Motivation

Real-time visual SLAM system aimed primarily for Augmented Reality (AR) applications

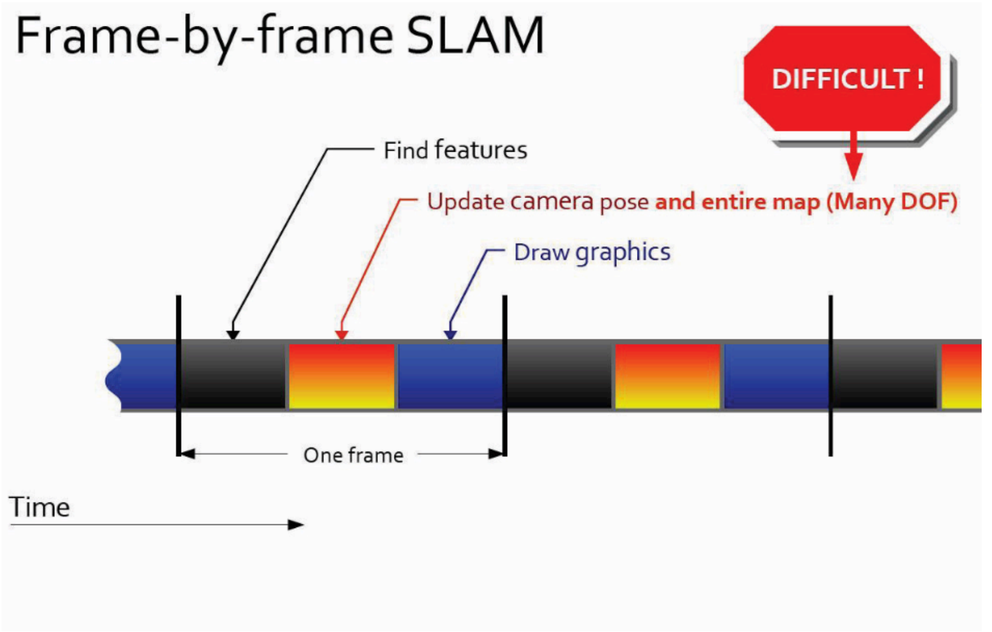
Existing state-of-the-art systems by Davison et al. [2] and Eade and Drummond [3] - adaptations of EKF-SLAM and FastSLAM 2.0 respectively

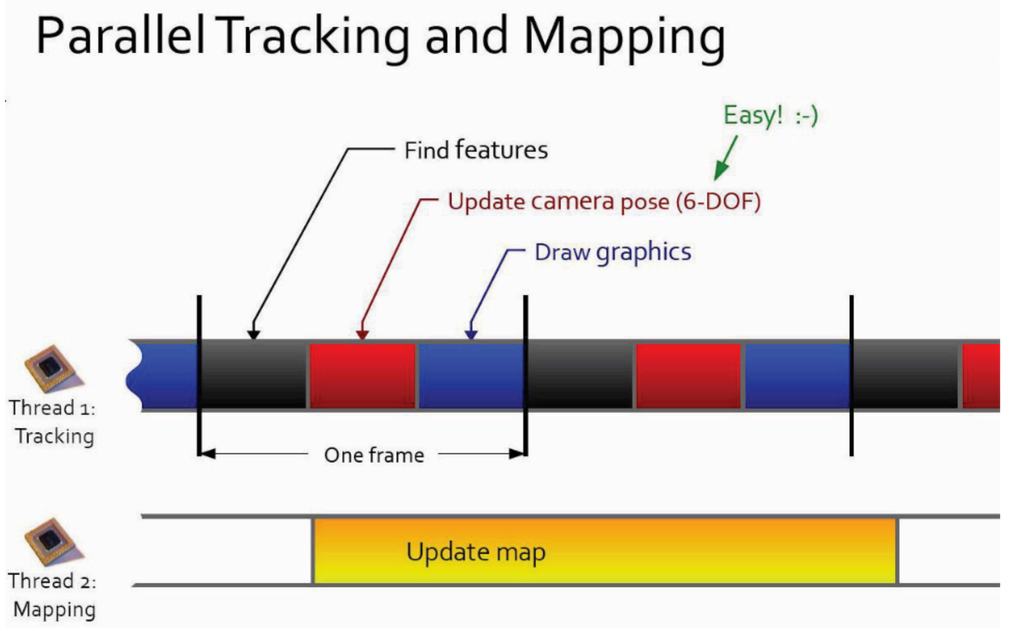
Both essentially "frame-by-frame" SLAM systems, with both current camera pose and landmark positions being updated together every frame

Not ideal for AR applications requiring tracking of a hand-held camera as opposed to a robot



https://www.doc.ic.ac.uk/~ab9515/ptam.html





https://www.doc.ic.ac.uk/~ab9515/ptam.html

Key Components

Tracking Thread

Responsible for estimating 6-DOF camera pose every frame

Must run at 30Hz

Focus on robustness and real-time performance

Mapping thread

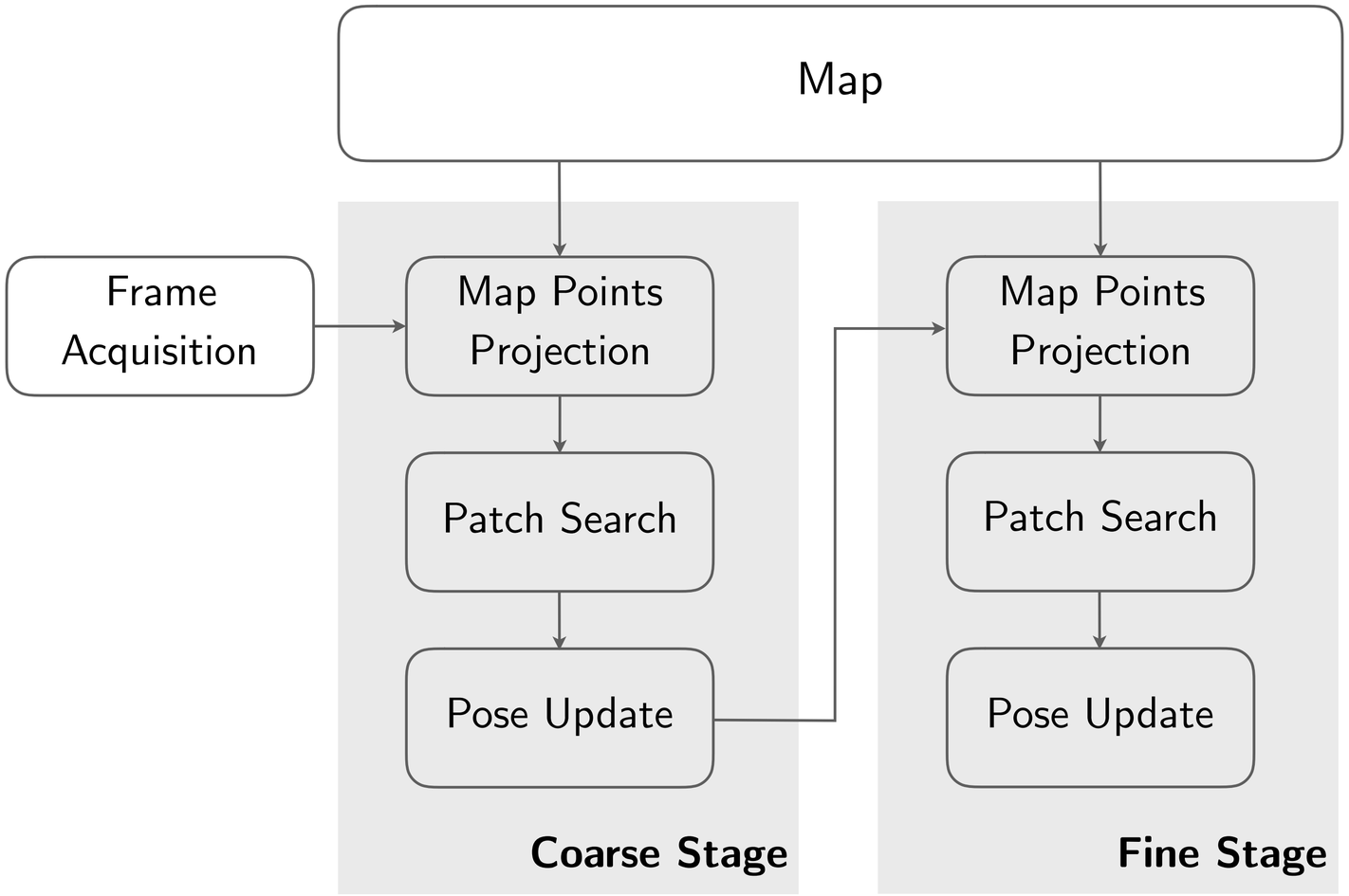
Responsible for mapping 3D points and refining past points and poses

Needs to update map every keyframe

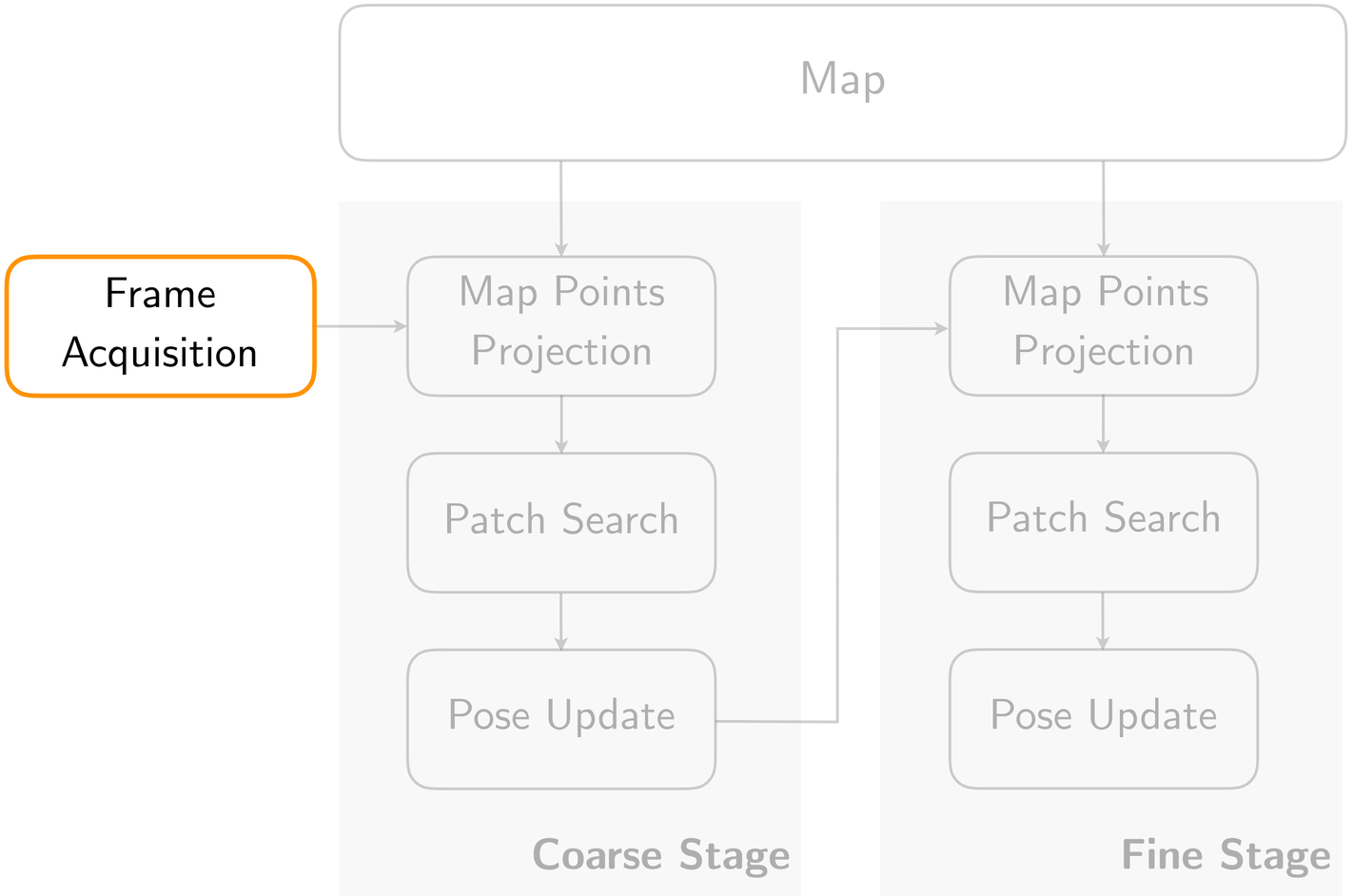
Focus on richness and accuracy

Tracking Thread

Tracking Components



Tracking Components



Frame Acquisition

Acquire most recent image frame

Construct an image pyramid with four levels

640 x 480 320 x 240 160 x 120 80 x 60

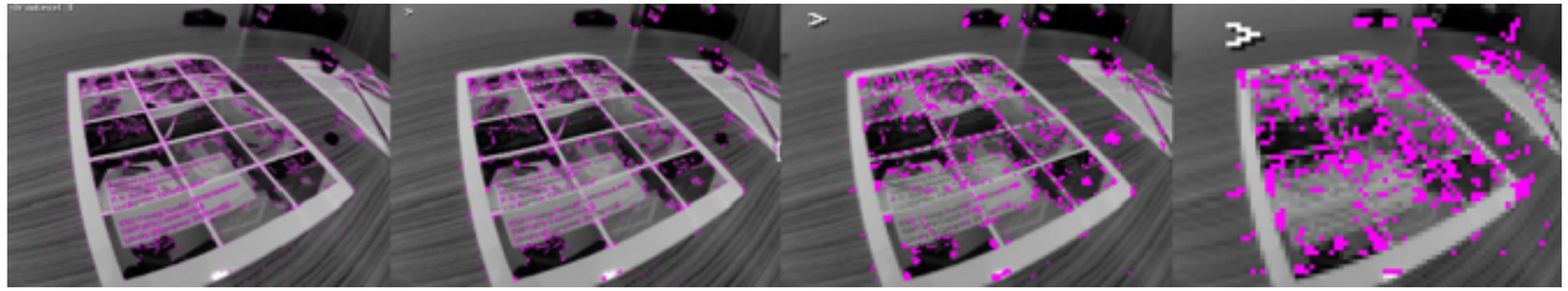
http://slideplayer.com/slide/4546633/

Frame Acquisition

Acquire most recent image frame

Construct an image pyramid with four levels

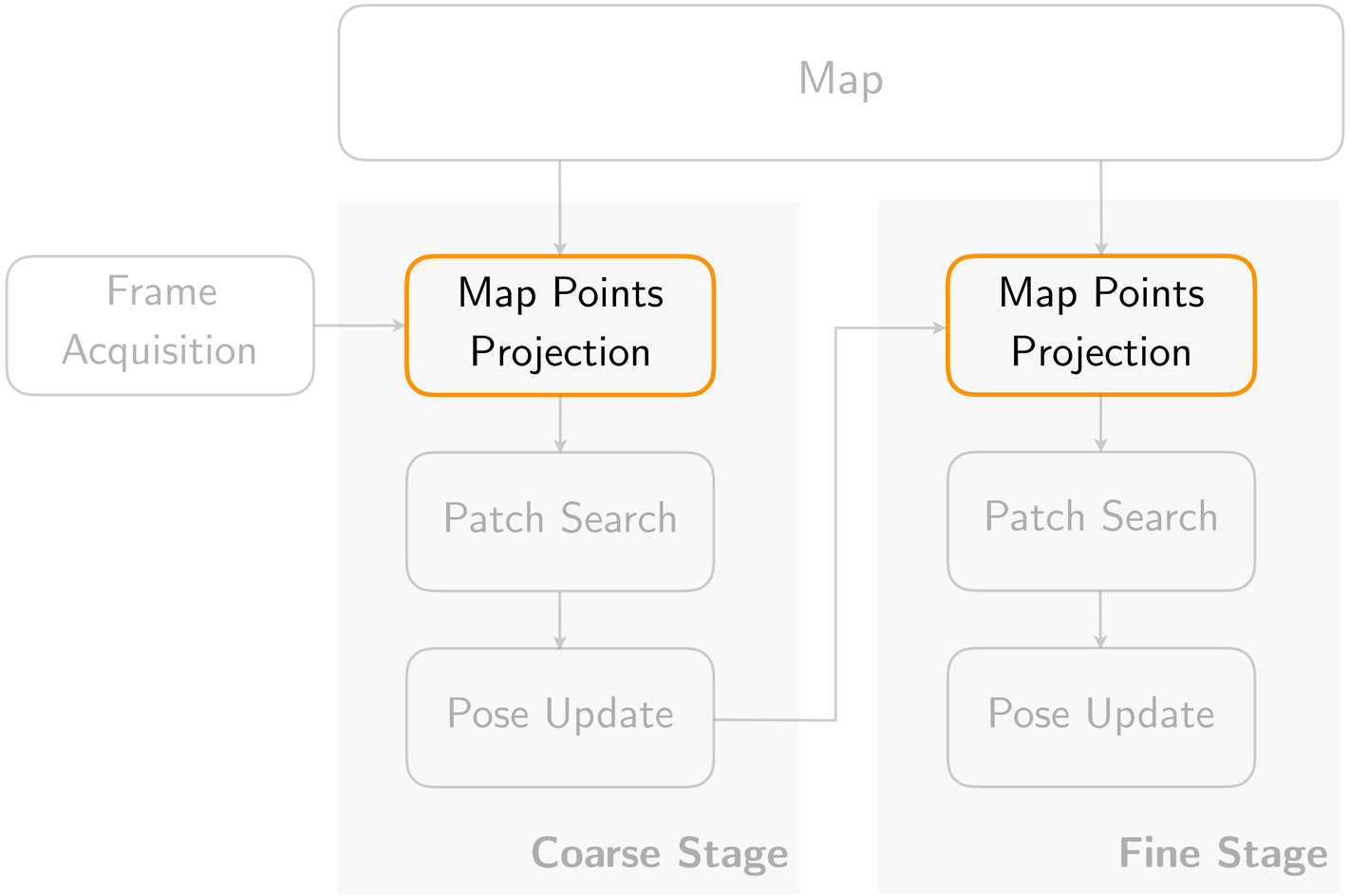
Detect FAST corners



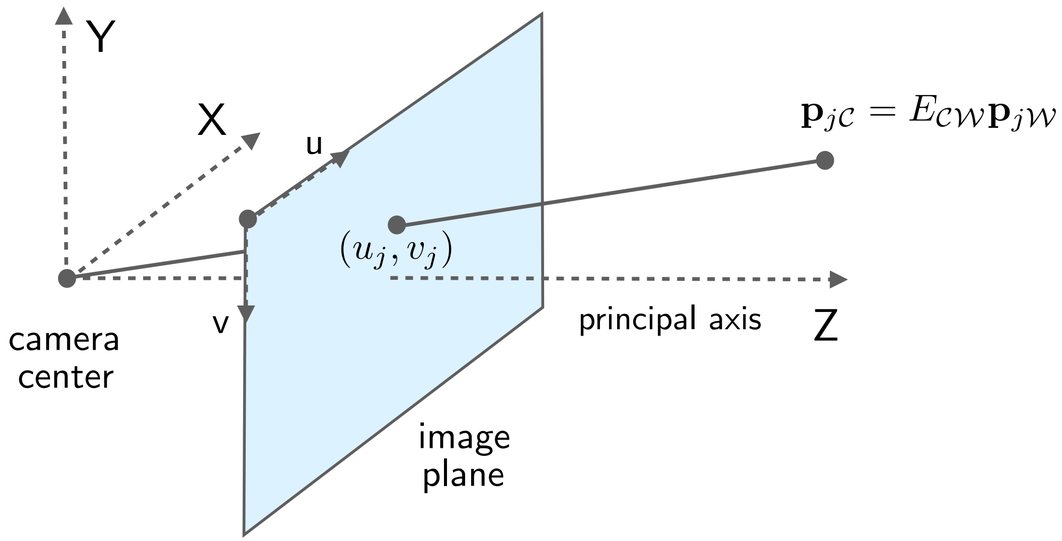
640 x 480 320 x 240 160 x 120 80 x 60

http://slideplayer.com/slide/4546633/

Tracking Components



Map Points Projection



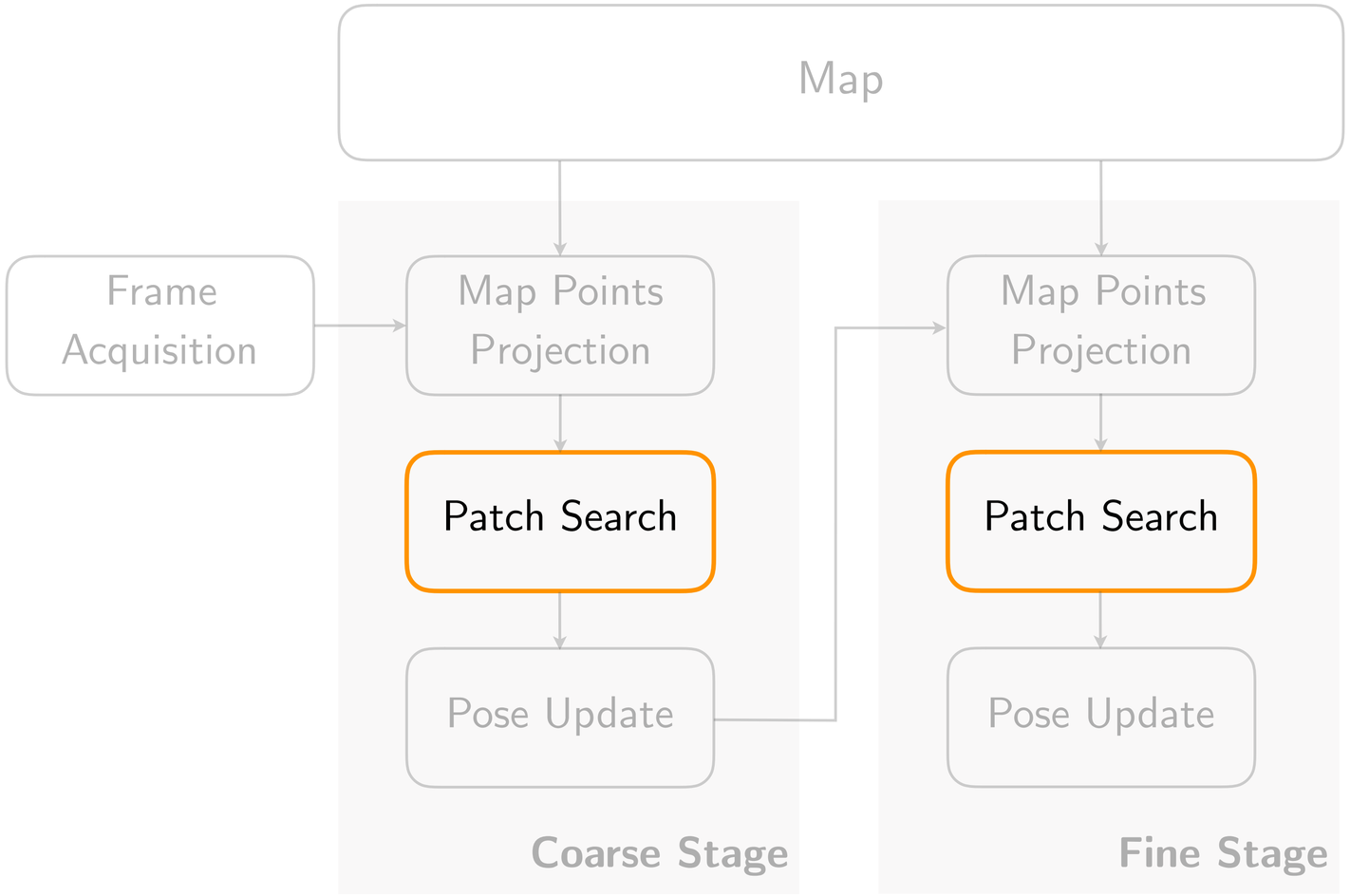
Project 3D map points $\mathbf{p}_{j\mathcal{W}}$ onto image plane of predicted camera pose $E_{\mathcal{C}\mathcal{W}}$,

$$\begin{pmatrix} u_j \\ v_j \end{pmatrix} = \mathrm{CamProj}\left(E_{\mathcal{C}\mathcal{W}}\mathbf{p}_{j\mathcal{W}}\right)$$

where,

$$\mathrm{CamProj}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \frac{r'}{r} \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

Tracking Components

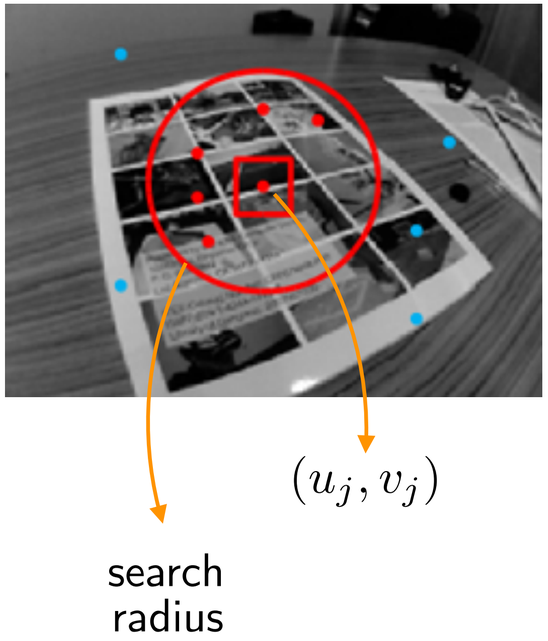


Patch Search

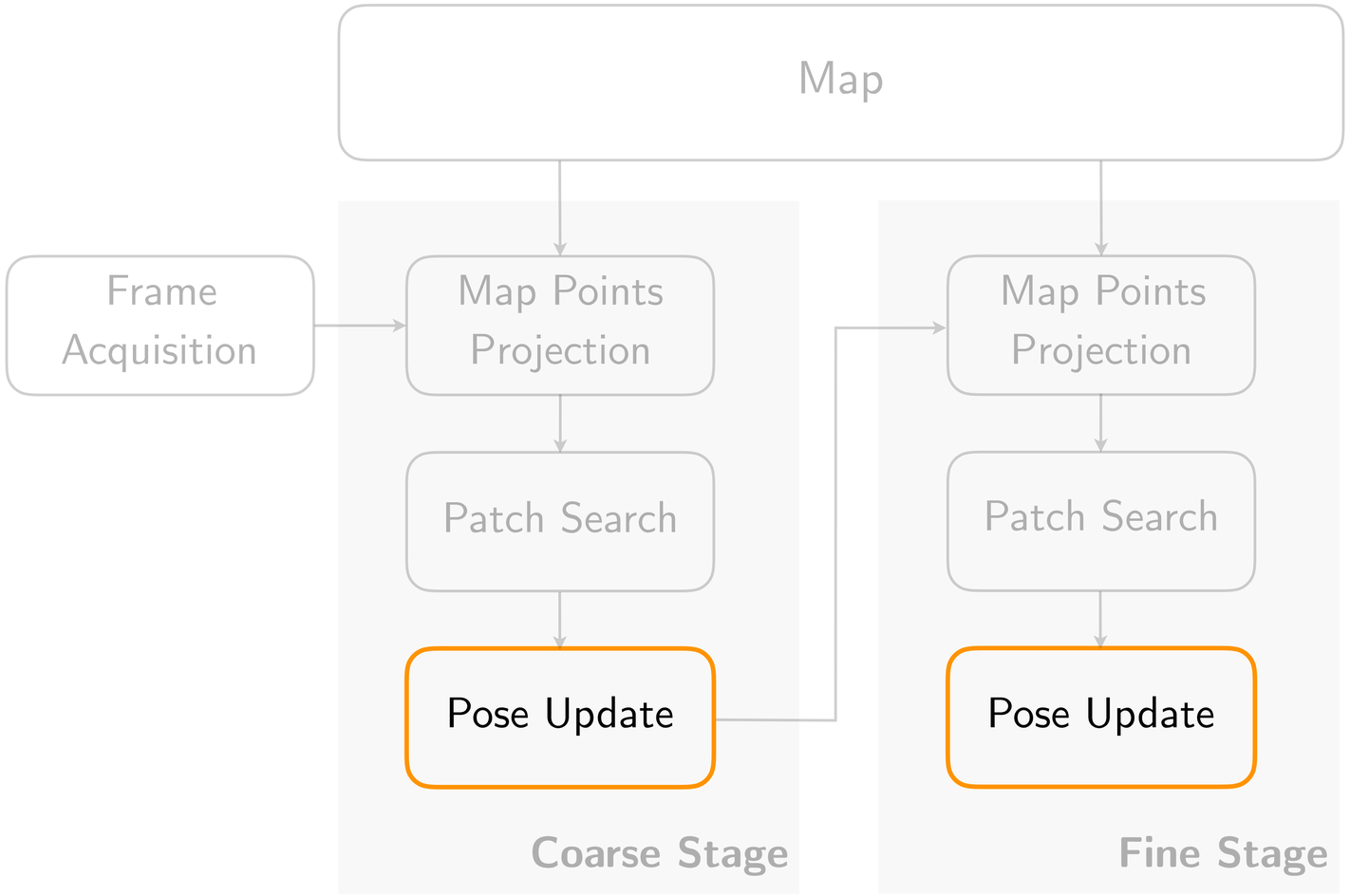
Generate 8x8 patch template around projected point (u_j, v_j)

Affine warping done to accommodate viewpoint changes between patch's first observation (source frame) and current camera position (target frame)

Search a fixed radius around projected point (u_j, v_j)
- Use zero-mean SSD
- Only search at FAST corner points



Tracking Components



Pose Update

Patch Search yields S successful patch matches with,

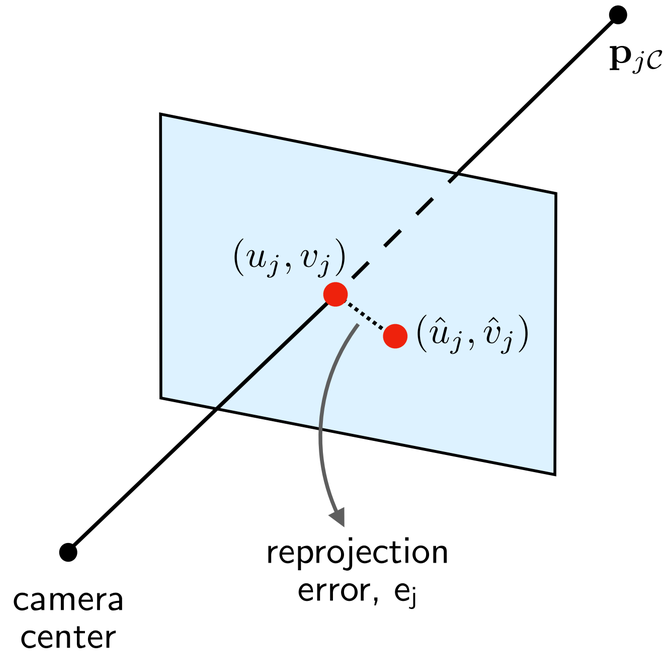
$(\hat{u}_j, \hat{v}_j)$ being matched patch centers corresponding to projected points (u_j, v_j)

Objective: Minimize reprojection error,

$$\mathbf{e}_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} - \mathrm{CamProj}\left(\exp\left(\boldsymbol{\mu}\right) E_{\mathcal{CW}} \mathbf{p}_{j\mathcal{W}}\right)$$

$$\boldsymbol{\mu}' = \mathrm{argmin}_{\boldsymbol{\mu}} \sum_{j \in S} \mathrm{Obj}\left(\frac{|\mathbf{e}_j|}{\sigma_j}, \sigma_T\right)$$

where, $\boldsymbol{\mu}$ is a 6-dim vector parameterizing camera motion using the exponential map.



Pose Update

Patch Search yields S successful patch matches with,

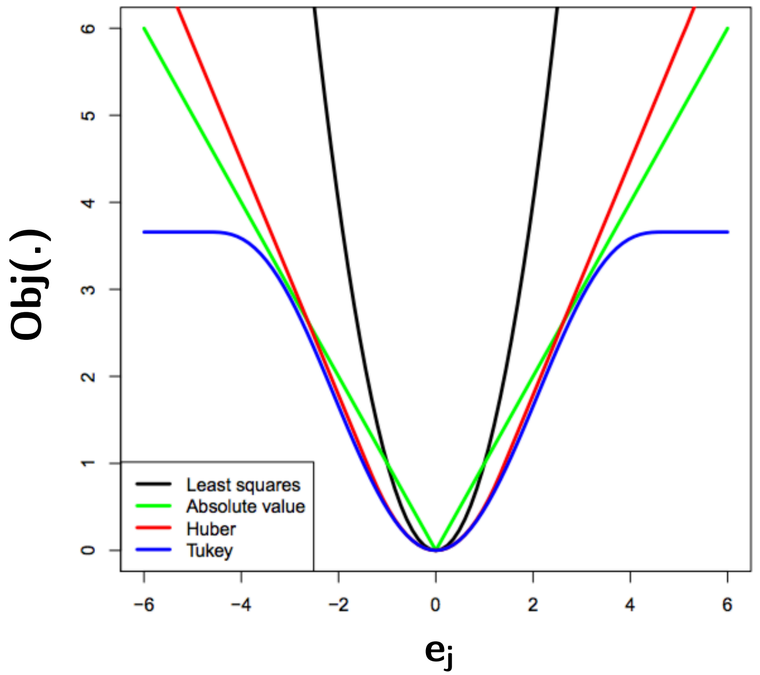
$(\hat{u}_j, \hat{v}_j)$ being matched patch centers corresponding to projected points (u_j, v_j)

Objective: Minimize reprojection error,

$$\mathbf{e}_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} - \mathrm{CamProj}\left(\exp\left(\boldsymbol{\mu}\right) E_{\mathcal{CW}} \mathbf{p}_{j\mathcal{W}}\right)$$

$$\boldsymbol{\mu}' = \mathrm{argmin}_{\boldsymbol{\mu}} \sum_{j \in S} \mathrm{Obj}\left(\frac{|\mathbf{e}_j|}{\sigma_j}, \sigma_T\right)$$

where, $\boldsymbol{\mu}$ is a 6-dim vector parameterizing camera motion using the exponential map.



https://web.as.uky.edu/statistics/users/pbreheny/764-F11/notes/12-1.pdf

Pose Update

Patch Search yields S successful patch matches with,

$(\hat{u}_j, \hat{v}_j)$ being matched patch centers corresponding to projected points (u_j, v_j)

Objective: Minimize reprojection error,

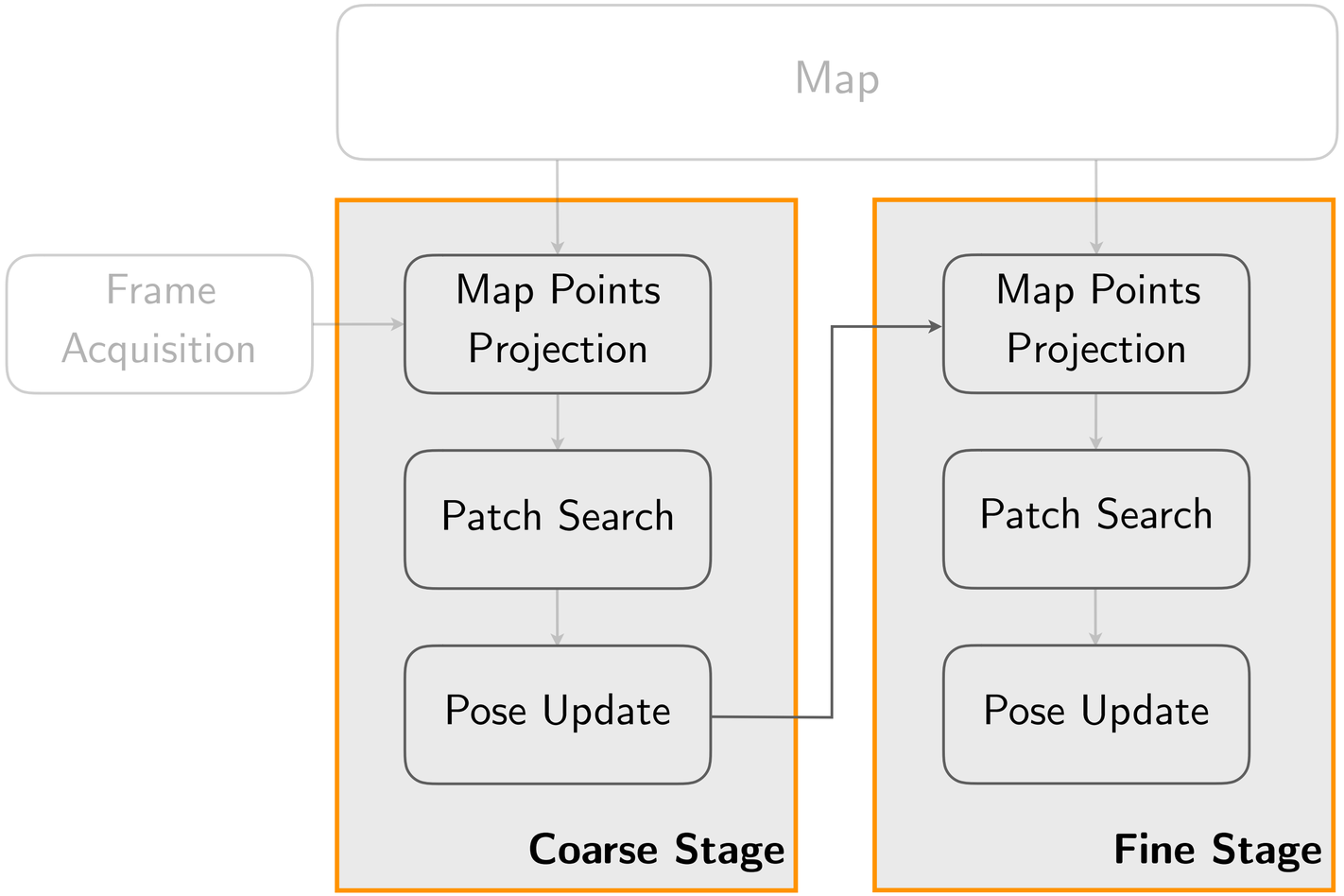
$$\mathbf{e}_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} - \mathrm{CamProj}\left(\exp\left(\boldsymbol{\mu}\right) E_{\mathcal{C}\mathcal{W}} \mathbf{p}_{j\mathcal{W}}\right)$$

$$\boldsymbol{\mu}' = \mathrm{argmin}_{\boldsymbol{\mu}} \sum_{j \in S} \mathrm{Obj}\left(\frac{|\mathbf{e}_j|}{\sigma_j}, \sigma_T\right)$$

Updates only current 6-dof camera pose: Cheap!

where, $\boldsymbol{\mu}$ is a 6-dim vector parameterizing camera motion using the exponential map.

Tracking Components



Points Projection, Patch Search, Pose Update

Repeat as two-stage coarse-to-fine process,

Coarse Stage

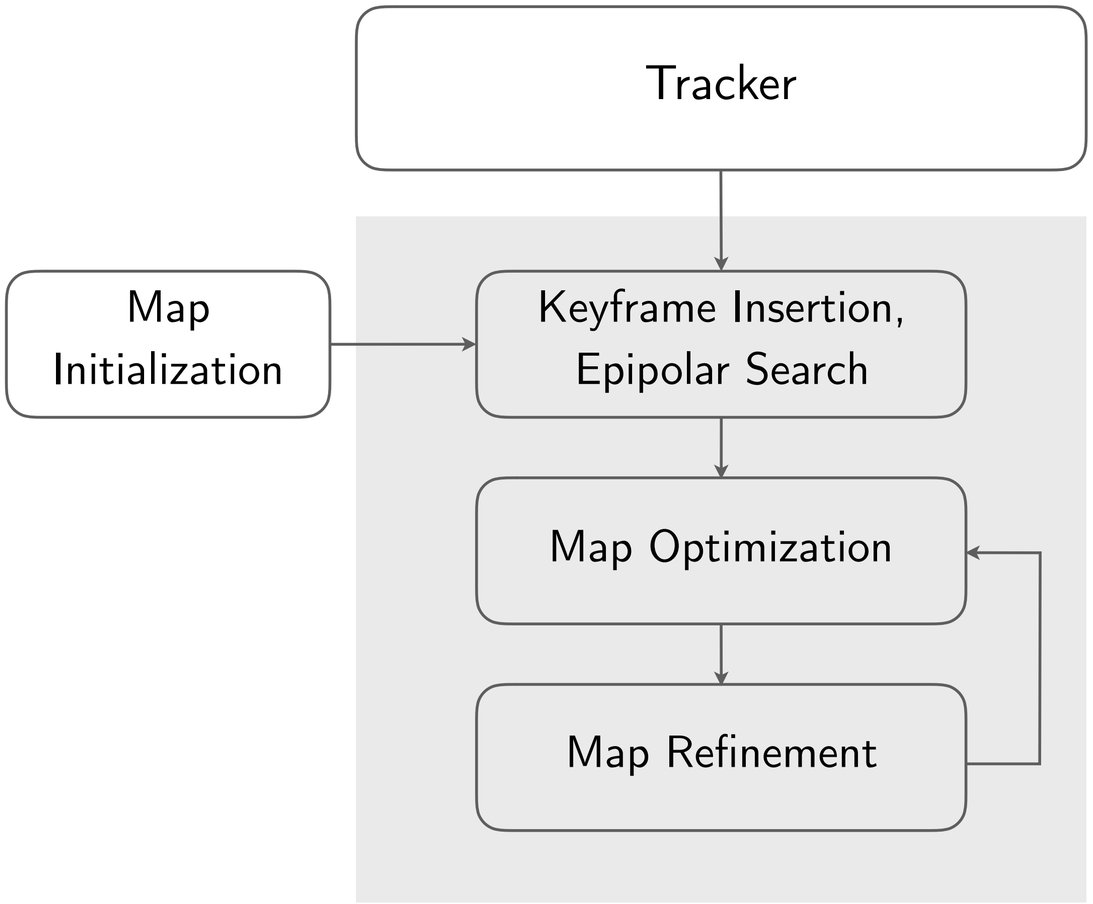
- Small number of points from coarser levels of the pyramid projected and searched in image
- Camera pose updated from these coarse matches

Fine Stage

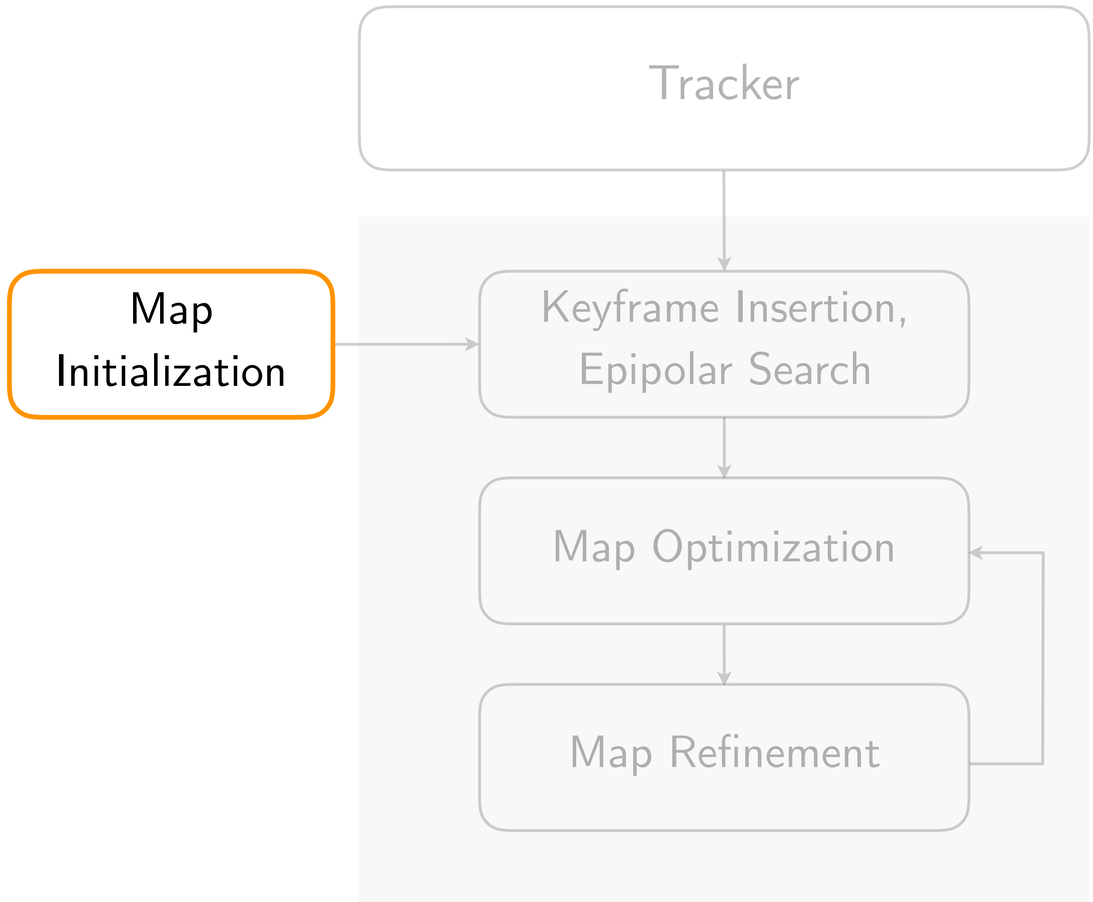
- Larger number of points from finer levels of the pyramid projected and searched in image
- Final pose estimate for the frame is computed from all matches found

Mapping Thread

Mapping Components



Mapping Components



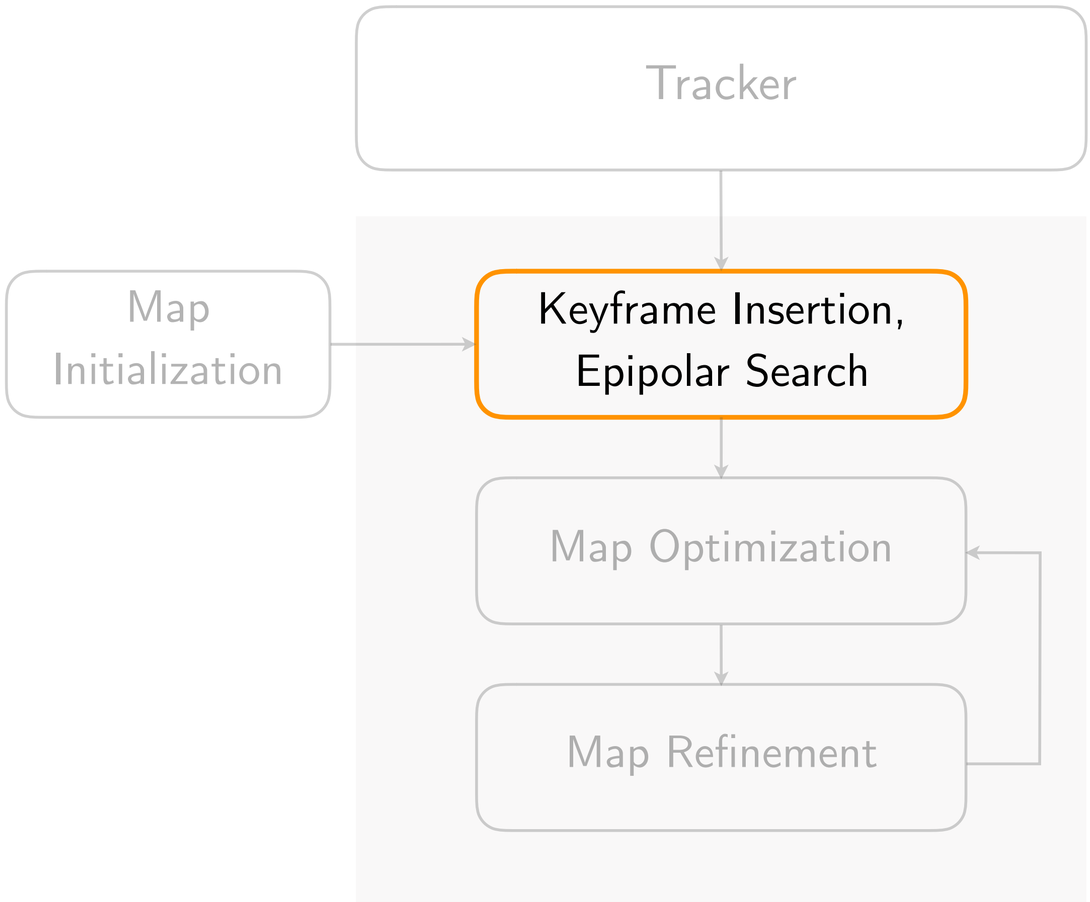
Map Initialization

User interactive initialization using feature correspondences from first two keyframes

Estimates essential matrix E_{12} and triangulates an initial map using five-point algorithm and RANSAC,

$q_{1j},\ q_{2j}$	pair of feature matches in keyframes 1,2
$q_{1j}^T E_{12} q_{2j} = 0$	solve eqn for E_{12}, min no. of matches needed is 5
$E_{12} = [t]_X R$	obtain (R,t) from essential matrix E_{12}

Mapping Components



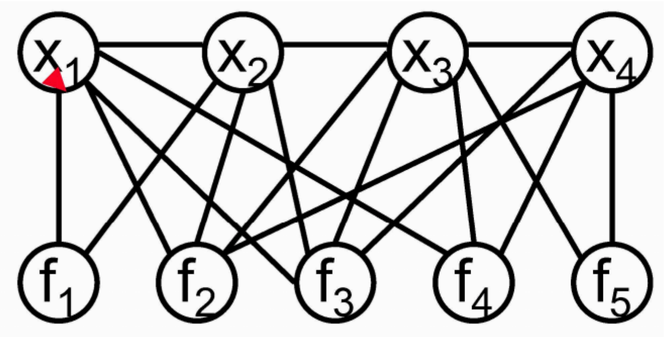
Keyframe Insertion, Epipolar Search

Keyframes Insertion only if,

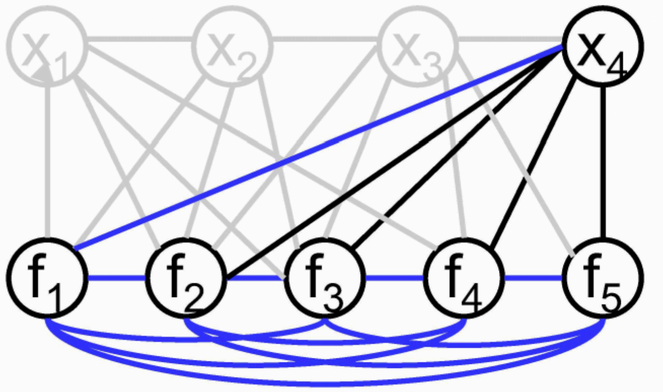
- Sufficient baseline with respect to already present keyframes
- Tracking quality is good

Epipolar Search for adding and refining 3D points in map,

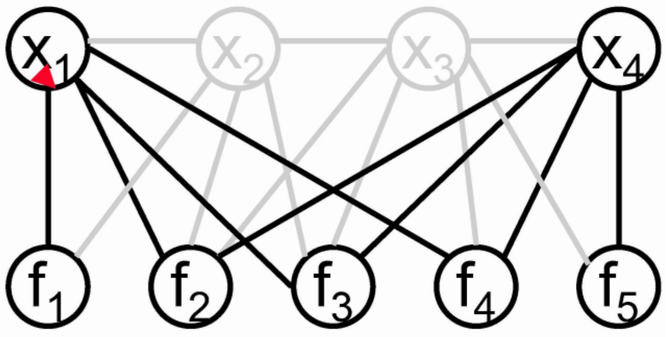
- Two-view triangulation of image features
- Closest keyframe (in terms of camera position) selected as second view



Original SLAM problem



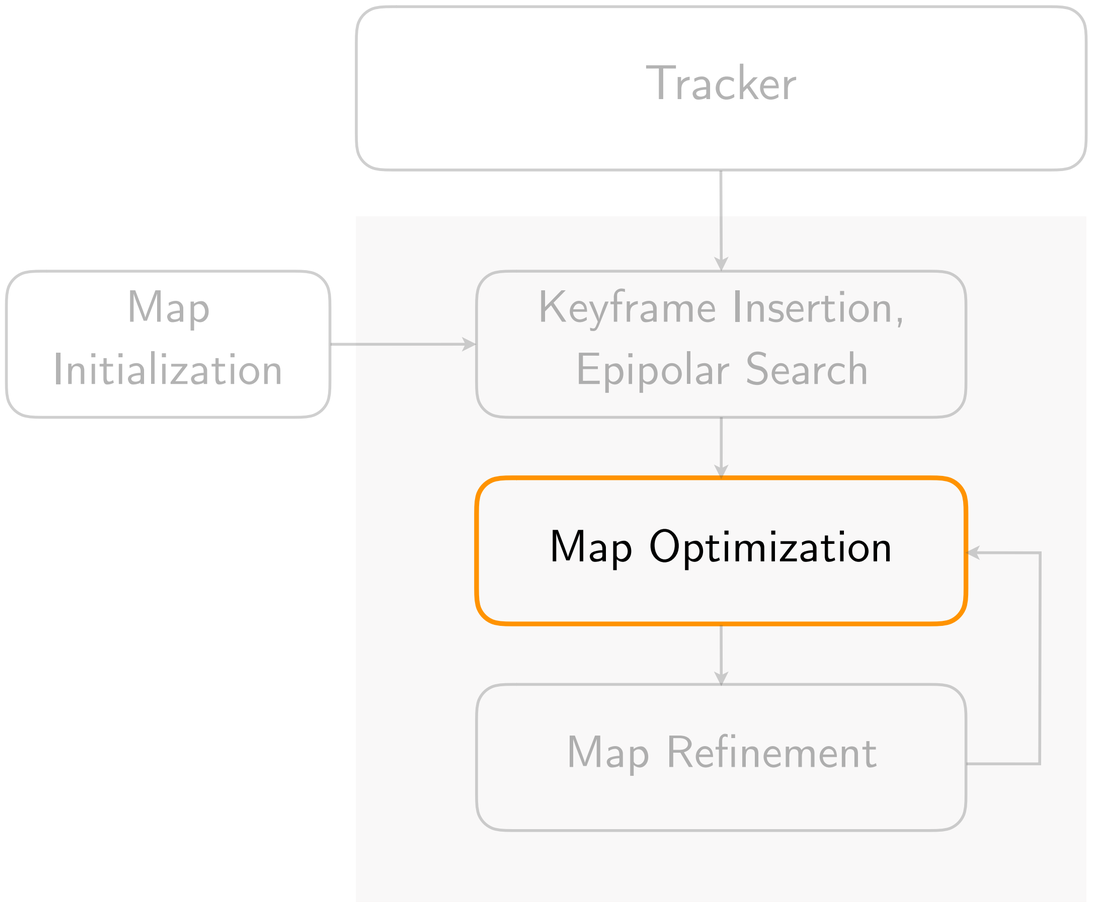
EKF approach



Keyframe approach (PTAM)

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

Mapping Components



Map Optimization

Global Bundle Adjustment

Iteratively adjust all M map points and N keyframe poses

$\boldsymbol{\mu} = \{\boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \ \dots \ \boldsymbol{\mu}_N\}$ camera pose motions for N keyframes

$\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \ \dots \ \mathbf{p}_M\}$ M 3D map points

Objective: Minimize reprojection error,

$$\{\{\boldsymbol{\mu}_2, \ \dots \ \boldsymbol{\mu}_N\}, \{\boldsymbol{p}_1, \ \dots \ \boldsymbol{p}_M\}\} = \text{argmin}_{\{\{\boldsymbol{\mu}\},\{\boldsymbol{p}\}\}} \sum_{i=1}^{N} \sum_{j \in S_i} \text{Obj}\left(\frac{|\mathbf{e}_{ji}|}{\sigma_{ji}}, \sigma_T\right)$$

where, $\mathbf{e}_{ji} = \begin{pmatrix} \hat{u}_{ji} \\ \hat{v}_{ji} \end{pmatrix} - \text{CamProj}\left(\exp\left(\boldsymbol{\mu}_i\right) E_{\mathcal{K}_i \mathcal{W}} \mathbf{p}_{j\mathcal{W}}\right)$

Map Optimization

Global Bundle Adjustment

Iteratively adjust all M map points and N keyframe poses

$\boldsymbol{\mu} = \{\boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \ \dots \ \boldsymbol{\mu}_N\}$ camera pose motions for N keyframes

$\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \ \dots \ \mathbf{p}_M\}$ M 3D map points

Objective: Minimize reprojection error,

$$\{\{\boldsymbol{\mu}_2, \ \dots \ \boldsymbol{\mu}_N\}, \{\boldsymbol{p}_1, \ \dots \ \boldsymbol{p}_M\}\} = \text{argmin}_{\{\{\boldsymbol{\mu}\},\{\boldsymbol{p}\}\}} \sum_{i=1}^{N} \sum_{j \in S_i} \text{Obj}\left(\frac{|\mathbf{e}_{ji}|}{\sigma_{ji}}, \sigma_T\right)$$

where, $\mathbf{e}_{ji} = \begin{pmatrix} \hat{u}_{ji} \\ \hat{v}_{ji} \end{pmatrix} - \text{CamProj}\left(\exp\left(\boldsymbol{\mu}_i\right) E_{\mathcal{K}_i \mathcal{W}} \mathbf{p}_{j\mathcal{W}}\right)$

Updates all camera poses and map points

Expensive!

Map Optimization

Local Bundle Adjustment

Iteratively adjust **subset** of all map points and keyframe poses

$$\{\{\boldsymbol{\mu}_{x\in X}\}\{\mathbf{p}_{z\in Z}\}\} = \text{argmin}_{\{\{\boldsymbol{\mu}\},\{\boldsymbol{p}\}\}} \sum_{i\in X\cup Y} \sum_{j\in Z\cap S_i} \text{Obj}\left(\frac{|\mathbf{e}_{ji}|}{\sigma_{ji}}, \sigma_T\right)$$

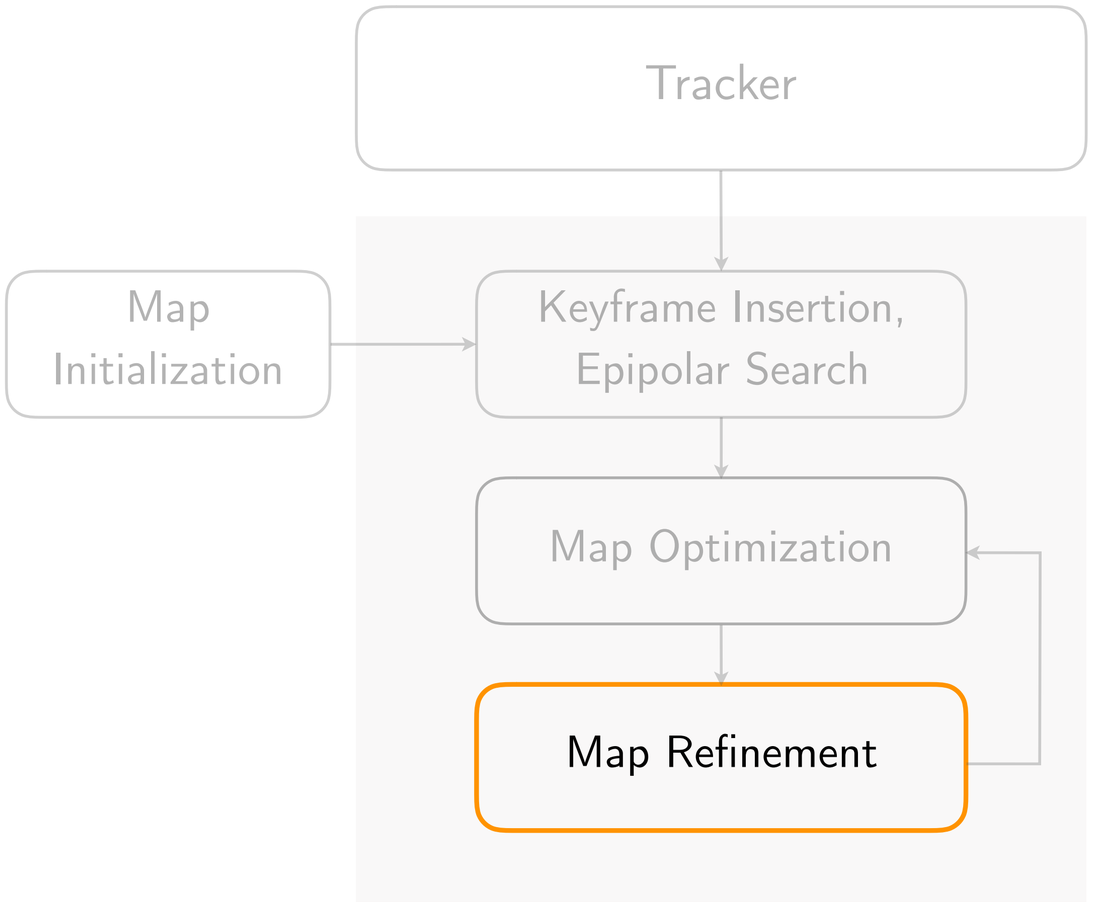
where,

X : set of keyframes to optimize (newest + four other keyframes nearest to it)

Z : all map points visible in any of these keyframes

Y: any keyframe for which a measurement of any point in Z has been made.

Mapping Components



Map Refinement

Refinement of Data Associations

- When bundle adjustment has converged and no new keyframes are added, mapping thread has idle time to improve the map

- Attempts at measuring new map features in all old key frames

- Reattempts outlier measurements

Results and Conclusions

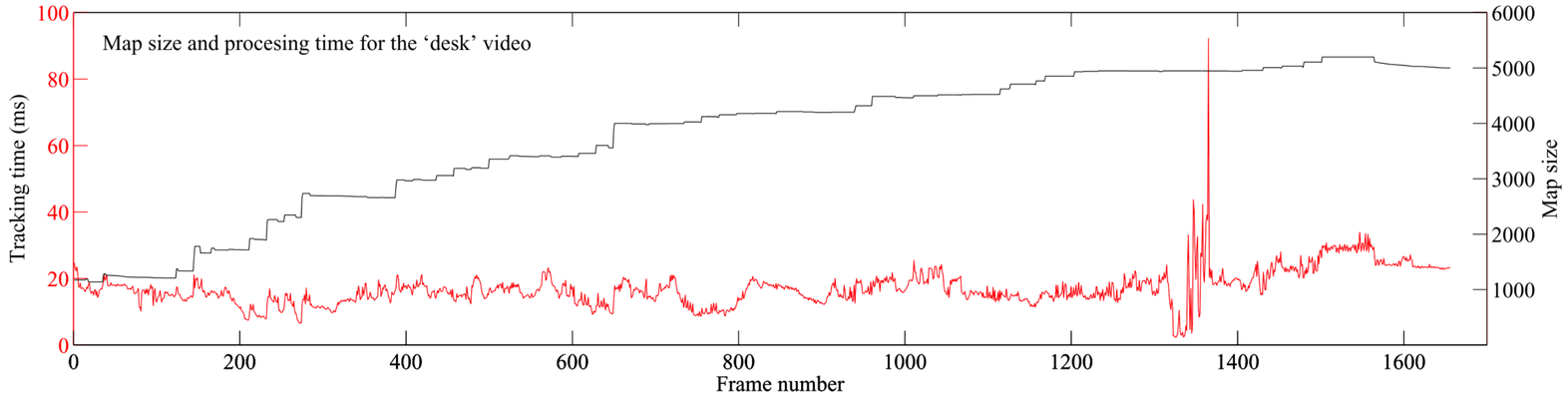
System Performance

Keyframe preparation	2.2 ms
Feature projection	3.5 ms
Patch search	9.8 ms
Iterative pose update	3.7 ms
Total	19.2 ms

Tracking timings for M=4000 (Tracking thread)

Keyframes	2-49	50-99	100-149
Local Bundle Adjustment	170ms	270ms	440ms
Global Bundle Adjustment	380ms	1.7s	6.9s

Bundle Adjustment timings (Mapping thread)



Map Size and Tracking timings

Parallel Tracking and Mapping for Small AR Workspaces

Extra video results made for ISMAR 2007 conference

Georg Klein and David Murray
Active Vision Laboratory
University of Oxford

https://www.youtube.com/watch?v=Y9HMn6bd-v8&

Conclusions

A real-time SLAM system falling under feature-based visual SLAM class of approaches

One of the first methods to use idea of parallel tracking and mapping. Ideas like key framing, parallel threads for tracking/mapping still used in modern feature-based visual SLAM systems (eg. ORB-SLAM)

Limitations,

- Restricted to small environments: no large-scale loop closing, keyframes grow unbounded
- Map initialization requires user interaction
- Designed for monocular cameras, can be brittle in scenarios like repeated textures

References

Papers

[1] Klein, Georg, and David Murray. "Parallel tracking and mapping for small AR workspaces." Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), 2007.

[2] A. Davison, I. Reid, N. D.Molton, and O. Stasse. MonoSLAM: Real- time single camera SLAM. to appear in IEEE Trans. Pattern Analysis and Machine Intelligence, 2007.

[3] E. Eade and T. Drummond. Scalable monocular slam. In Proc. IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR '06), pages 469–476, New York, NY, 2006.

Questions?