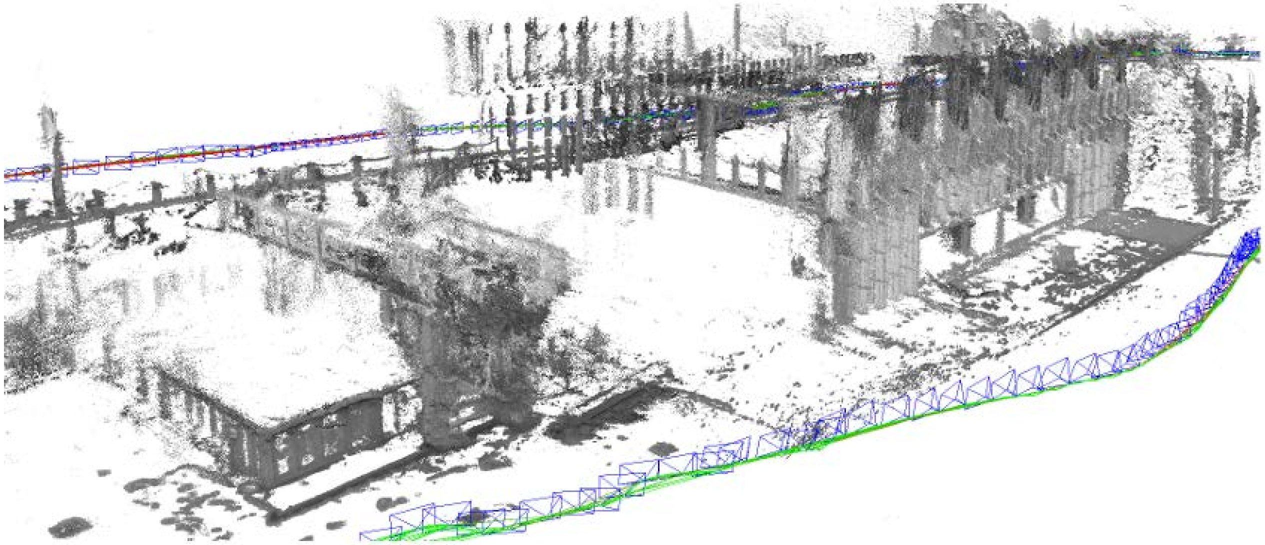
LSD-SLAM: Large-Scale Direct Monocular SLAM

Jakob Engel, Thomas Schöps, Daniel Cremers
Technical University Munich

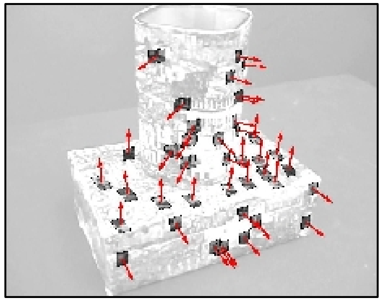


Monocular Video → **Camera Motion** and **Scene Geometry**

Live Operation

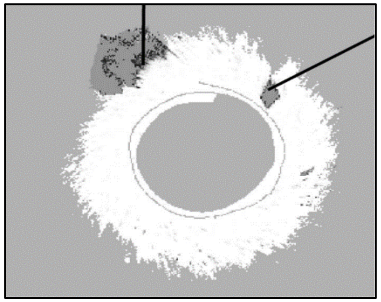
real-time operation on laptop (no GPU)

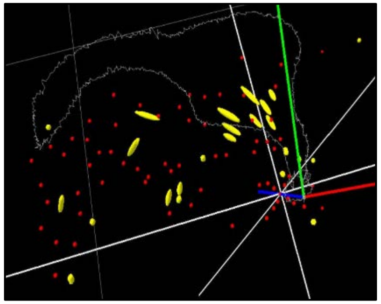
(Some) Related Work



Structure from Motion Causally Integrated Over Time.
Chiuso, Favaro, Jin, Soatto; PAMI ‘02

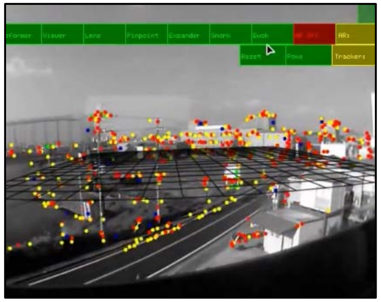
Visual Odometry.
Nistér, Naroditsky, Bergen; CVPR ‘04

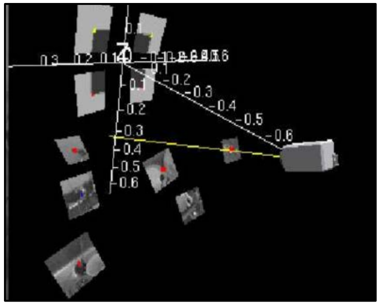




Scalable monocular SLAM.
Eade, Drummond; CVPR ‘06

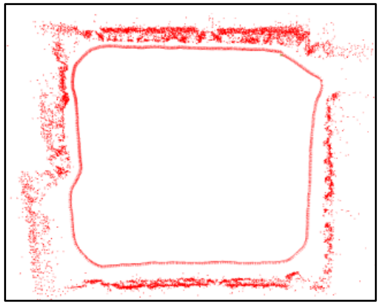
Parallel Tracking and Mapping for Small AR Workspaces. *Klein, Murray*; ISMAR ‘07

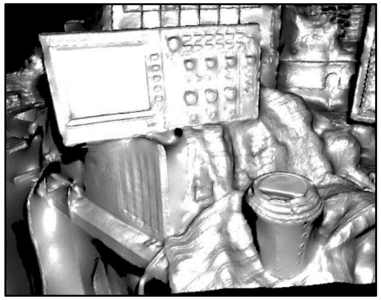




MonoSLAM: Real-time single camera SLAM.
Davison, Reid, Molton, Stasse; PAMI ‘07

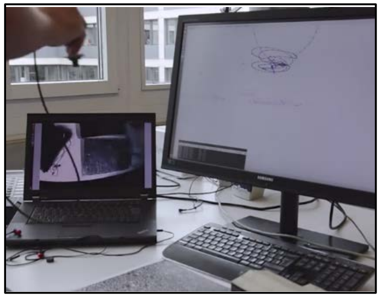
Scale Drift-Aware Large Scale Monocular SLAM.
Strasdat, Montiel, Davison; RSS ‘10





DTAM: Dense Tracking and Mapping in Real-Time.
Newcombe, Lovegrove, Davison; ICCV ‘11

SVO: Fast Semi-Direct Monocular Visual Odometry.
Forster, Pizzoli, Scaramuzza; ICRA ‘14



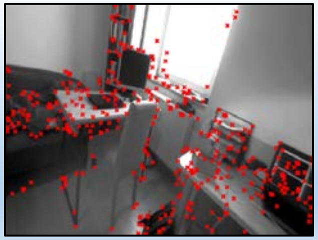
LSD-SLAM: what‘s new?

Keypoint-Based

Input Images

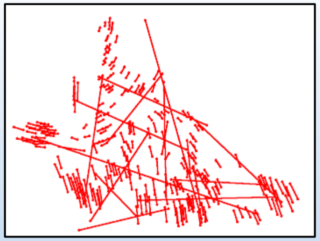
Extract & Match Features

(SIFT / SURF / BRIEF /...)

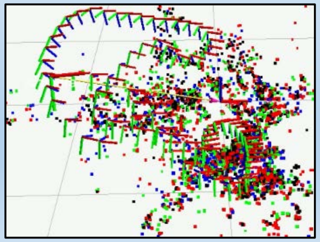


abstract images to feature observations

Track:
min. **reprojection** error
(point distances)



Map:
est. **feature-parameters**
(3D points / normals)

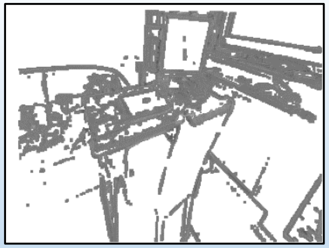


Direct (LSD-SLAM)

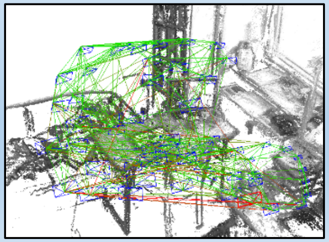
Input Images

keep full image

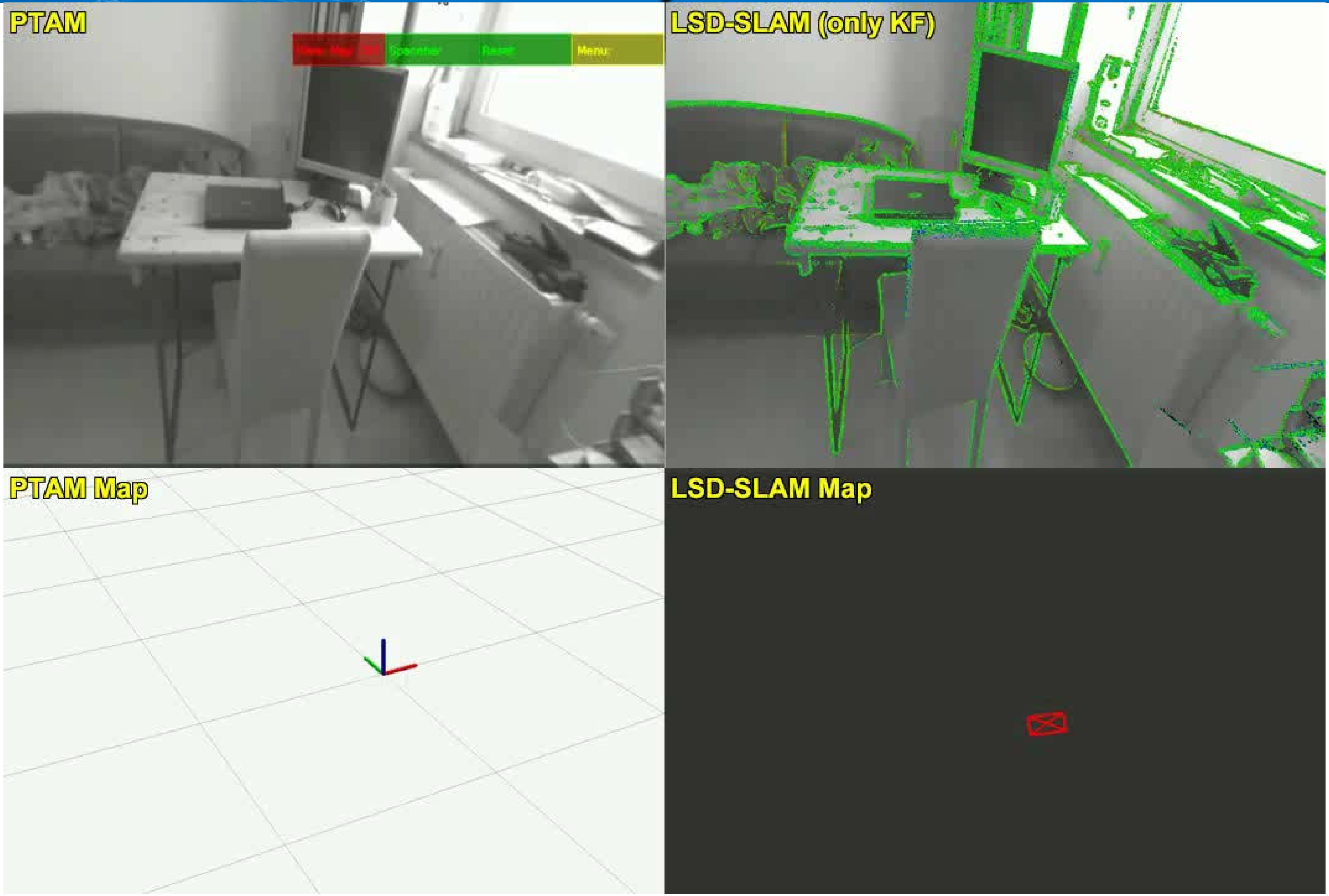
Track:
min. **photometric** error
(intensity difference)



Map:
est. **per-pixel depth**
(semi-dense depth map)



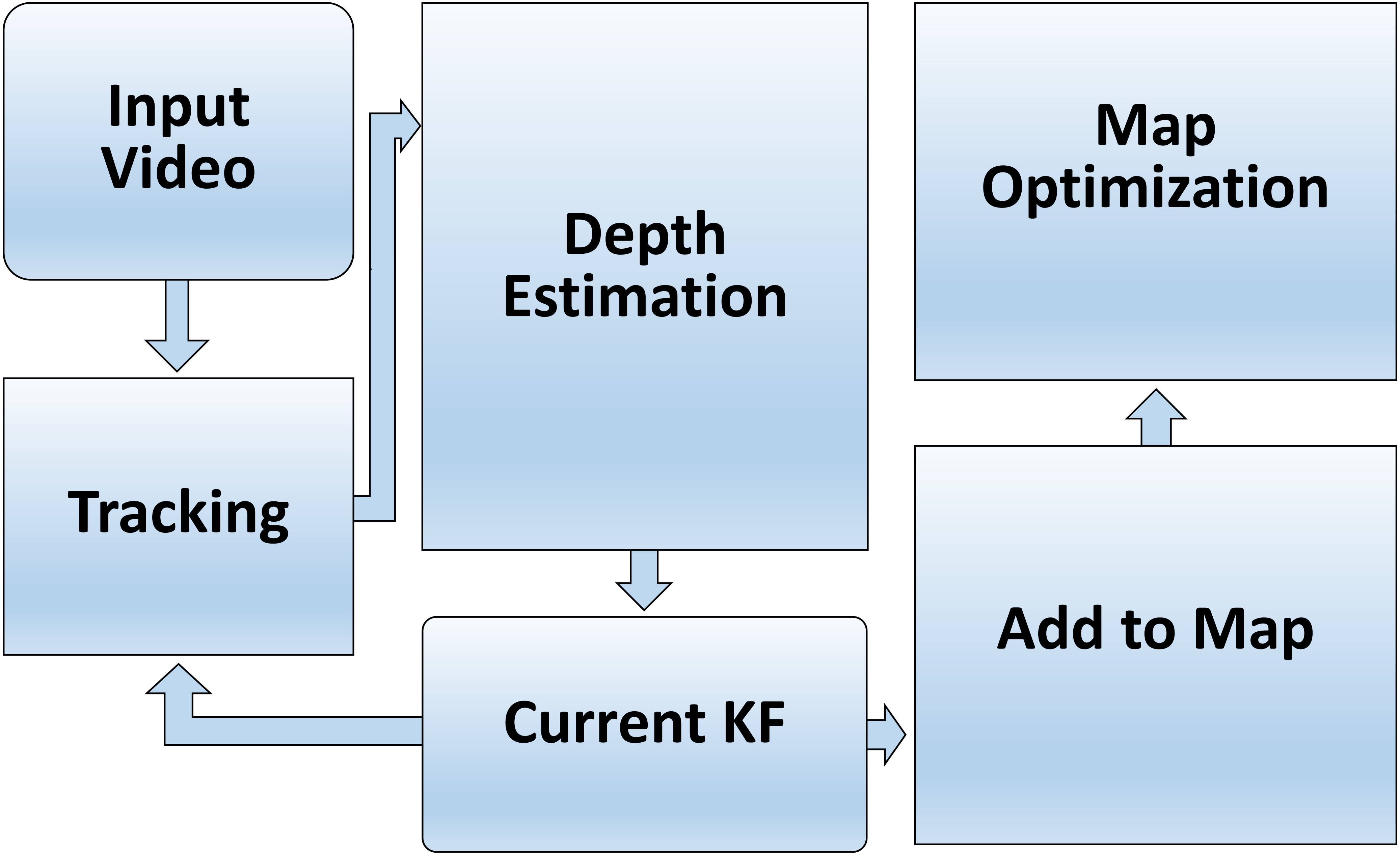
...and why do that?



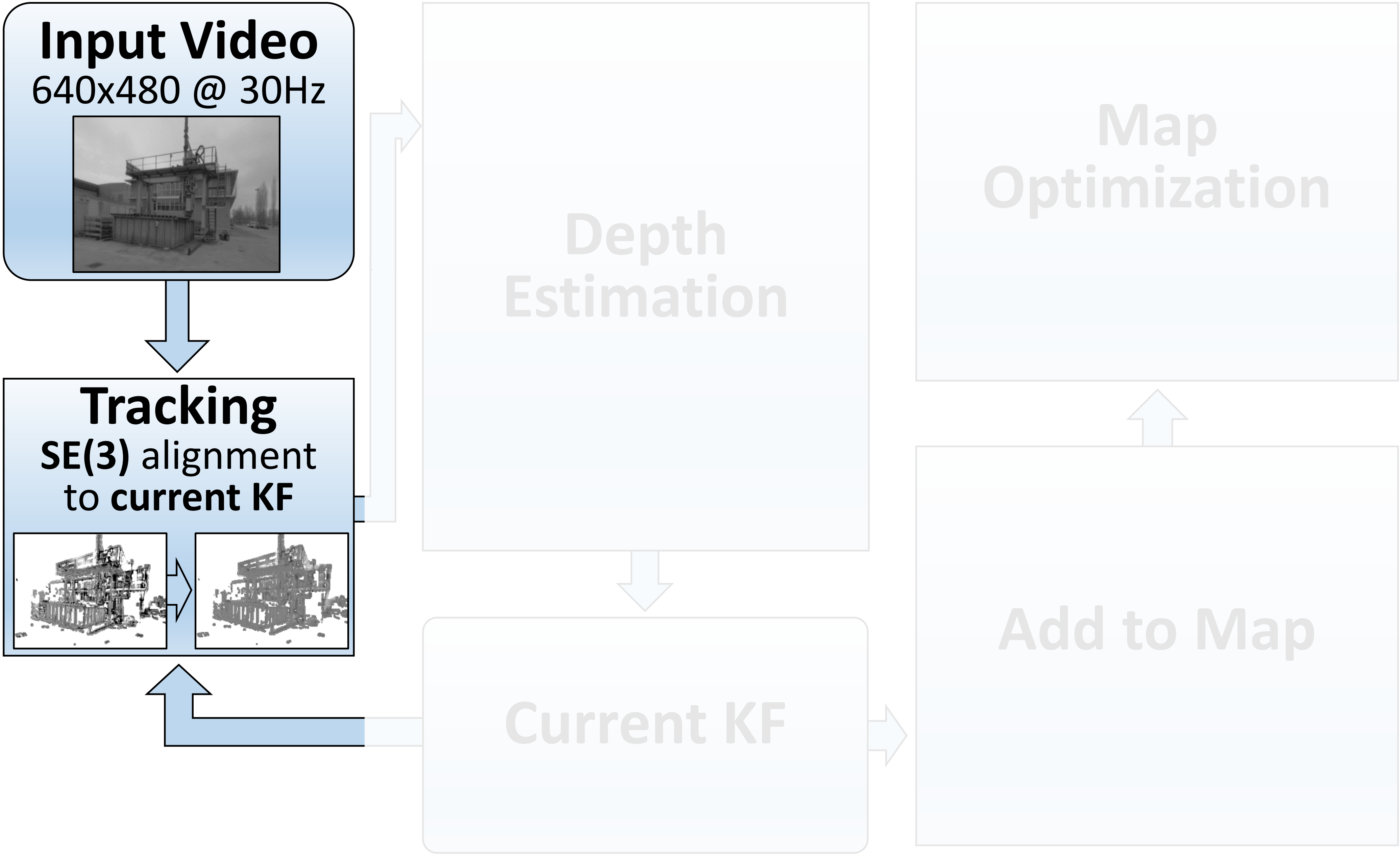
can only use & reconstruct corners

can use & reconstruct whole image

Overview


Input Video
Tracking
Depth Estimation
Current KF
Add to Map
Map Optimization

Overview



Input Video
640x480 @ 30Hz

Tracking
SE(3) alignment
to **current KF**

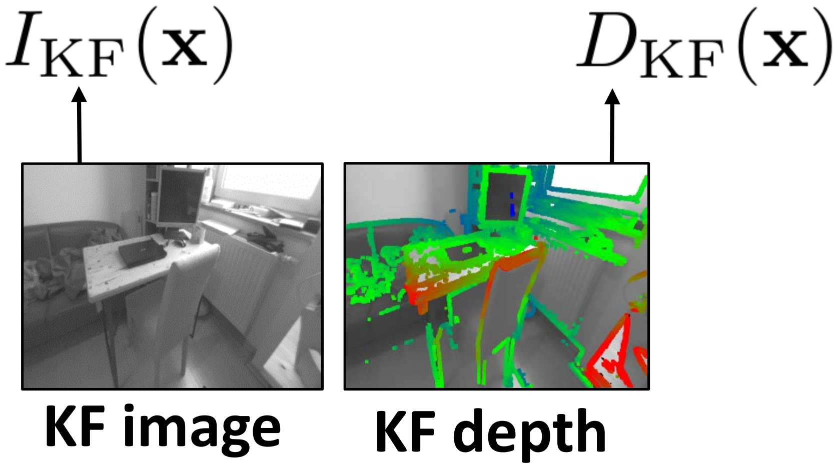
Depth Estimation

Current KF

Map Optimization

Add to Map

Tracking



$I_{\mathrm{KF}}(\mathbf{x})$

$D_{\mathrm{KF}}(\mathbf{x})$

KF image

KF depth

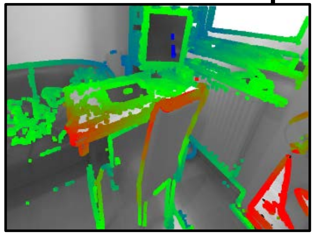
Tracking

$$\boldsymbol{\xi}$$

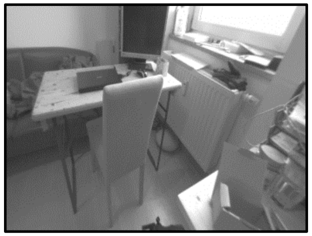
Camera Pose
in $\mathfrak{se}(3)$

$$I_{\mathrm{KF}}(\mathbf{x}) - \underbrace{I(\omega(\mathbf{x}, D_{\mathrm{KF}}(\mathbf{x}), \boldsymbol{\xi}))}$$

KF image



KF depth



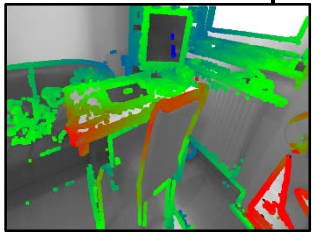
back-warped
new frame

Tracking

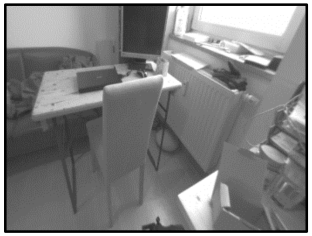
$$E(\boldsymbol{\xi}) = \sum_{\mathbf{x}\in\Omega_{\mathrm{KF}}} \Big(I_{\mathrm{KF}}(\mathbf{x}) - \underbrace{I(\omega(\mathbf{x}, D_{\mathrm{KF}}(\mathbf{x}), \boldsymbol{\xi}))}\Big)^2 =: \|\mathbf{r}(\boldsymbol{\xi})\|_2^2$$

Camera Pose in $\mathfrak{se}(3)$

KF image



KF depth



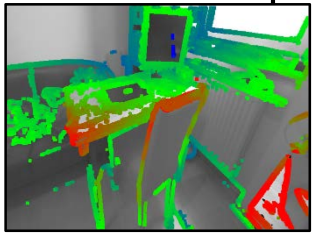
back-warped new frame

Tracking

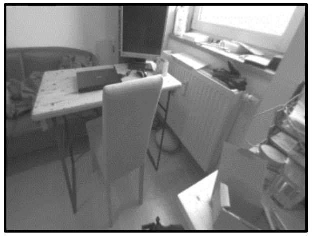
$$E(\boldsymbol{\xi}) = \sum_{\mathbf{x}\in\Omega_{\mathrm{KF}}} \Big(I_{\mathrm{KF}}(\mathbf{x}) - \underbrace{I(\omega(\mathbf{x}, D_{\mathrm{KF}}(\mathbf{x}), \boldsymbol{\xi}))}\Big)^2 =: \|\mathbf{r}(\boldsymbol{\xi})\|_2^2$$

Camera Pose in $\mathfrak{se}(3)$

KF image



KF depth



back-warped new frame

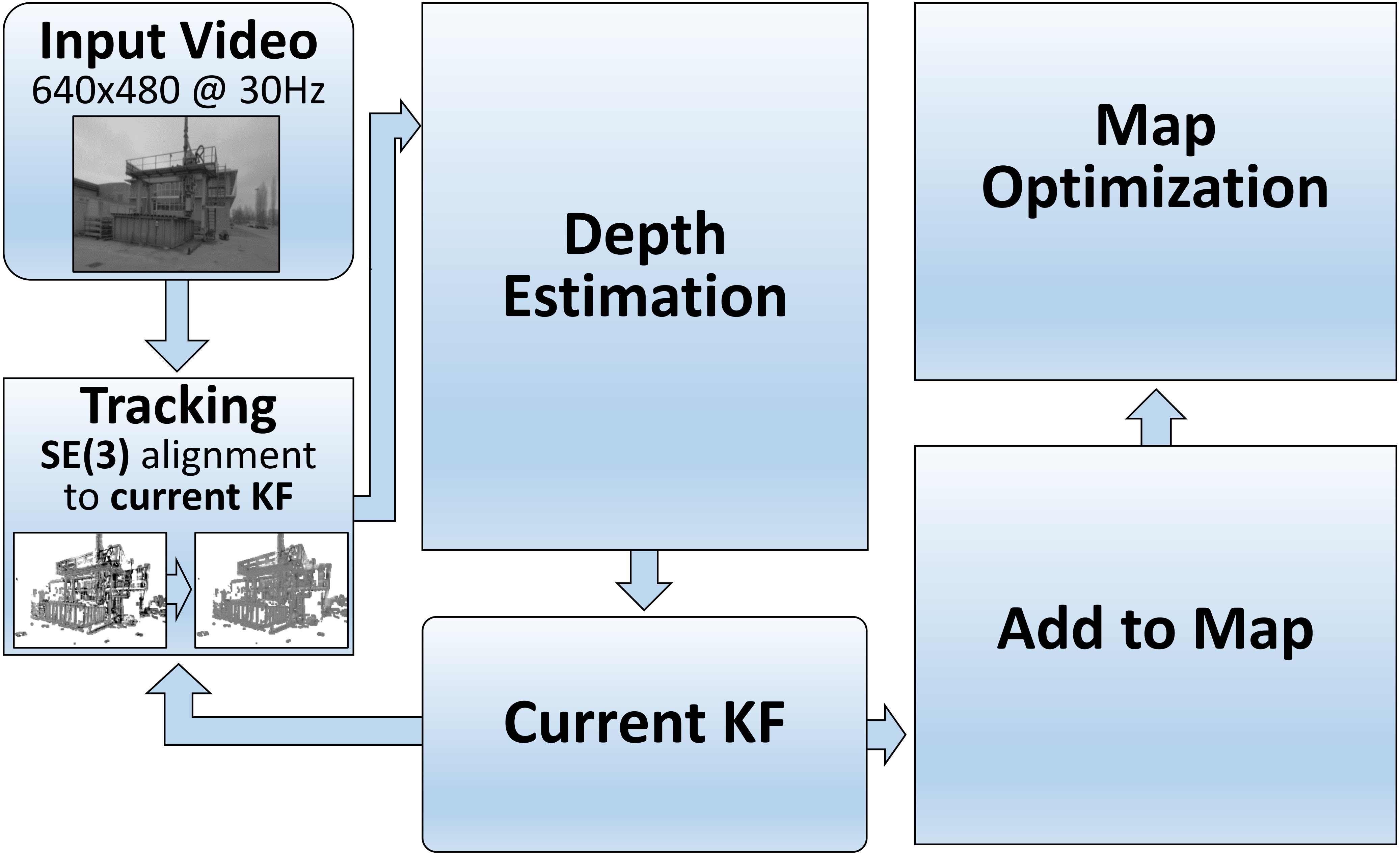
- ➢ minimize using **Gauss-Newton** Algorithm (≈ forward-compositional Lucas-Kanade)

Tracking

- **multi-resolution** (track large motions)
- **Huber norm instead of L2** (outliers & occlusions)
- **statistical normalization** (respect depth- and pixel-noise)

single core timings:
320x240: 5-10ms
640x480: 20-30ms

Overview



Input Video
640x480 @ 30Hz

Tracking
SE(3) alignment
to **current KF**

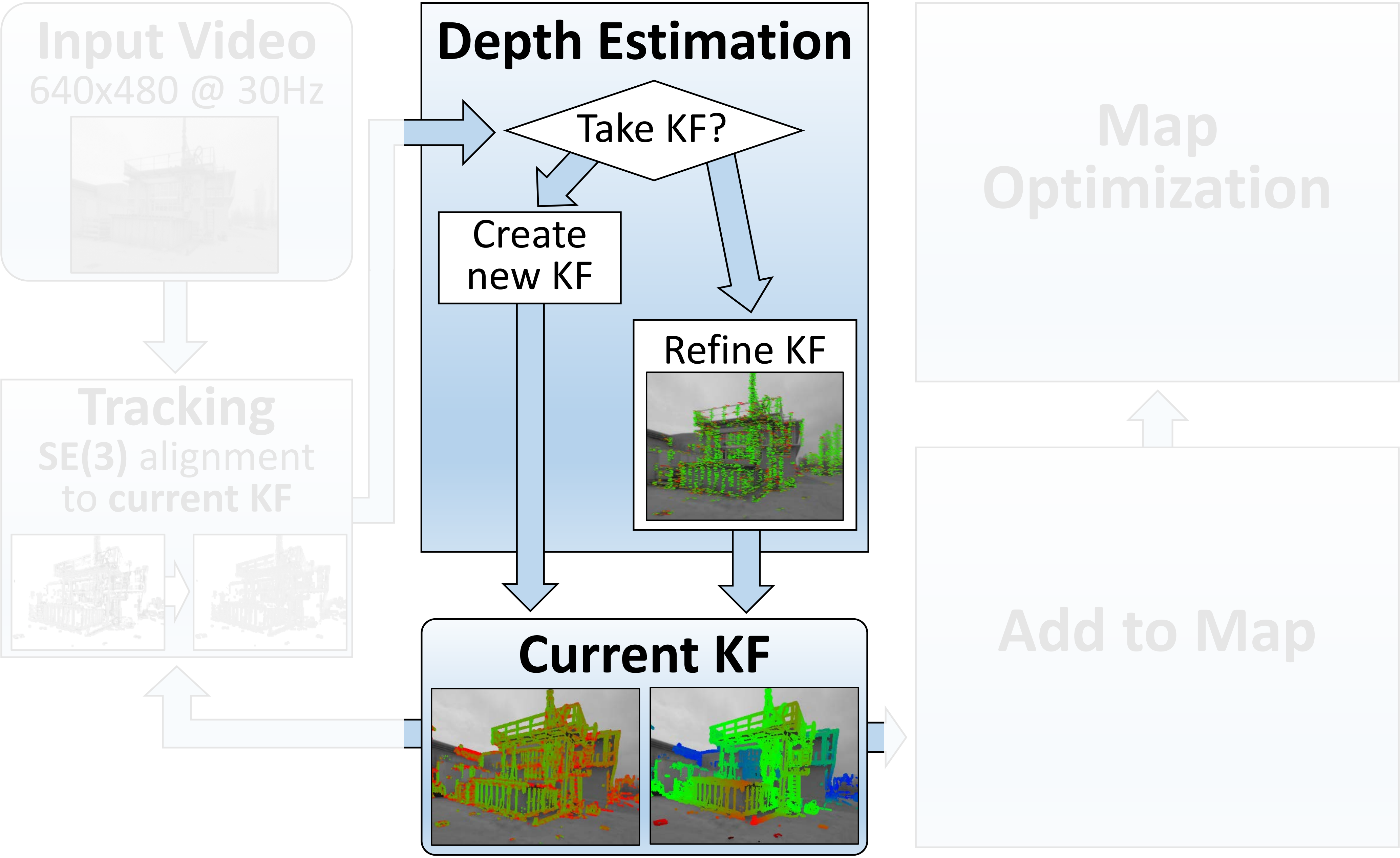
Depth Estimation

Current KF

Map Optimization

Add to Map

Overview



Input Video

640x480 @ 30Hz

Tracking

SE(3) alignment to current KF

Depth Estimation

Take KF?

Create new KF

Refine KF

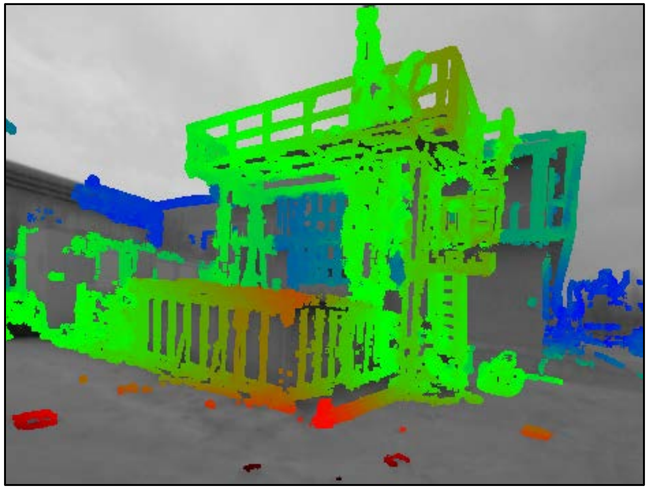
Current KF

Map Optimization

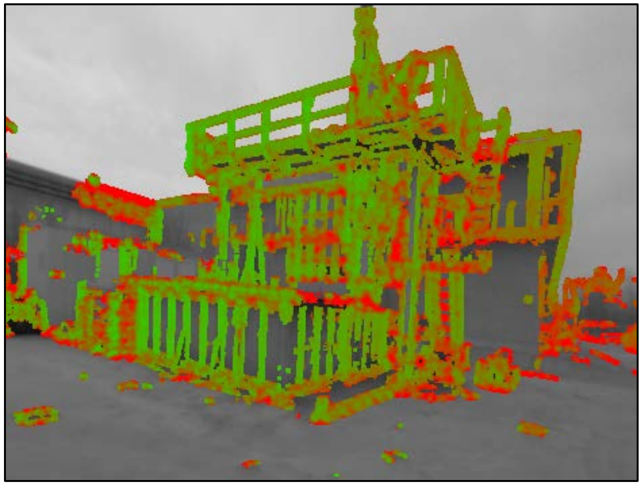
Add to Map

Depth Estimation

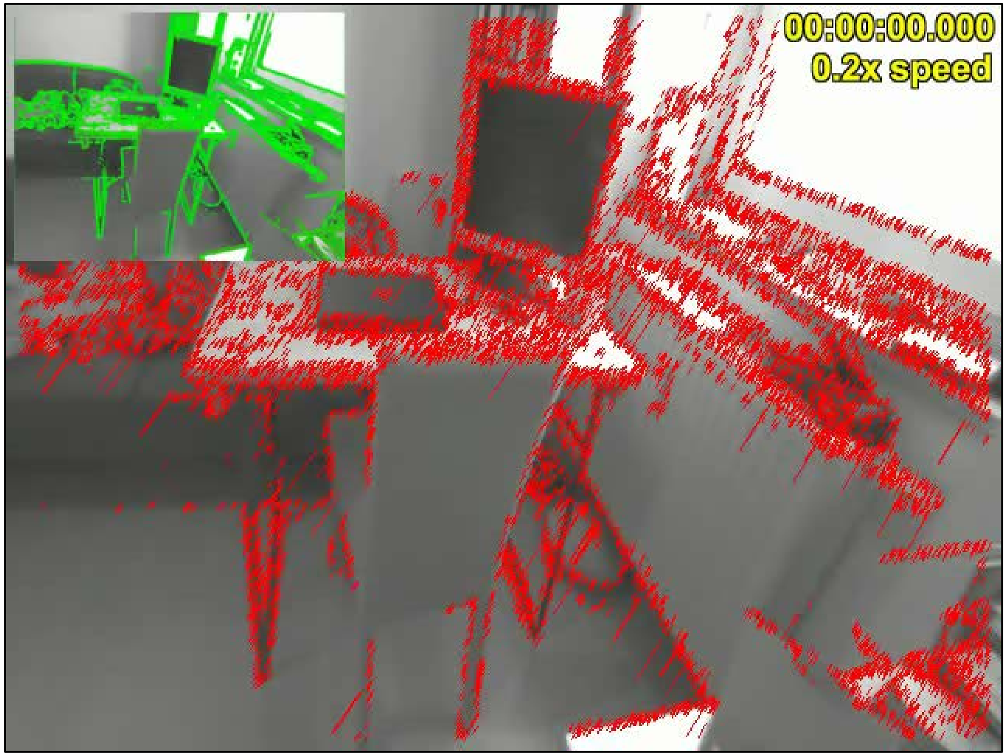
image



inverse depth



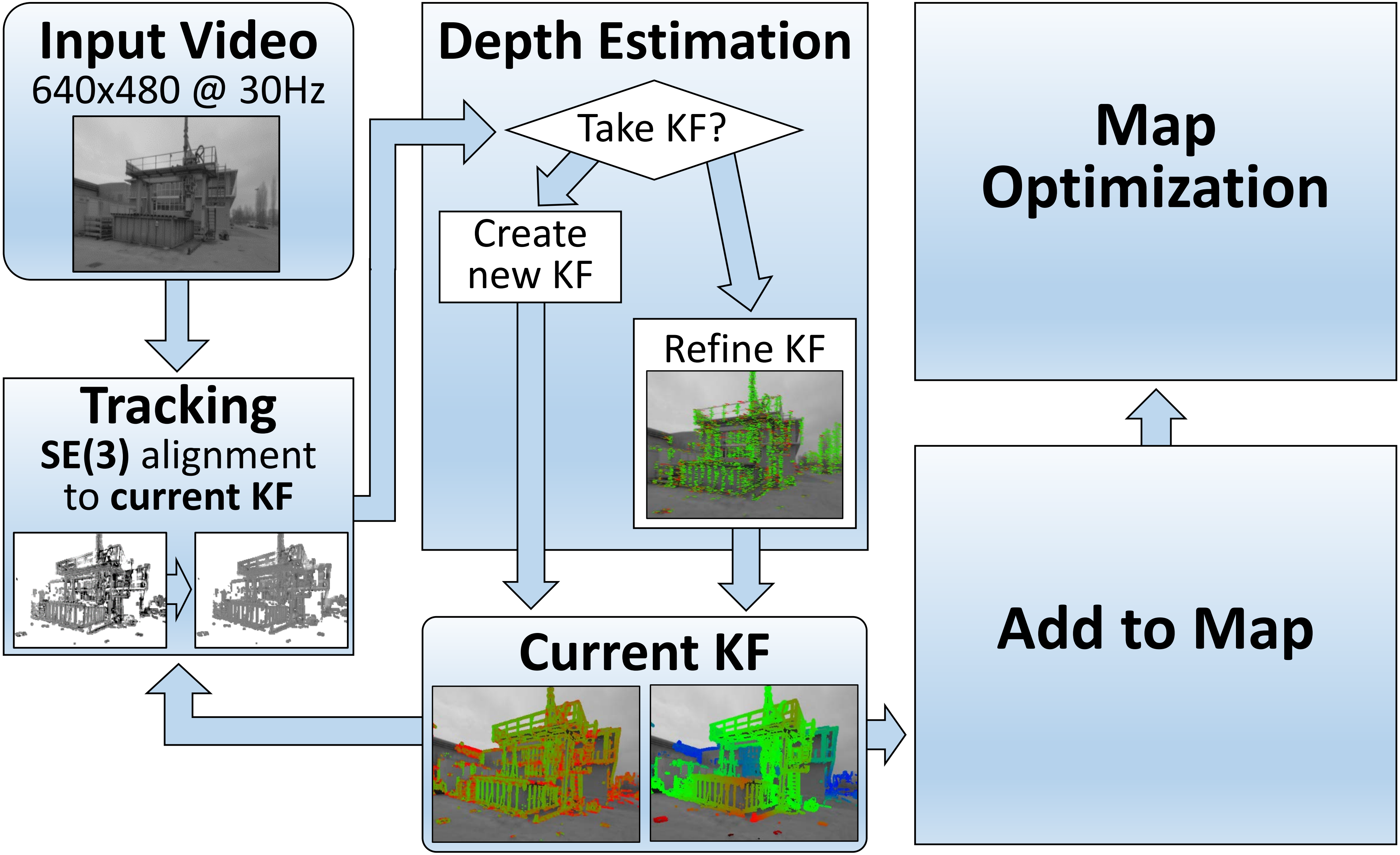
inverse depth variance



- **pixelwise filtering** (exploit video)
  small-baseline → large baseline
- **information selection**
  „only do stereo if sufficient information gain“
- **edge-preserving smoothing**
- **distance-based KF selection**

[Engel, Sturm, Cremers; ICCV ´13]

Overview



Input Video

640x480 @ 30Hz

Tracking

SE(3) alignment to **current KF**

Depth Estimation

Take KF?

Create new KF

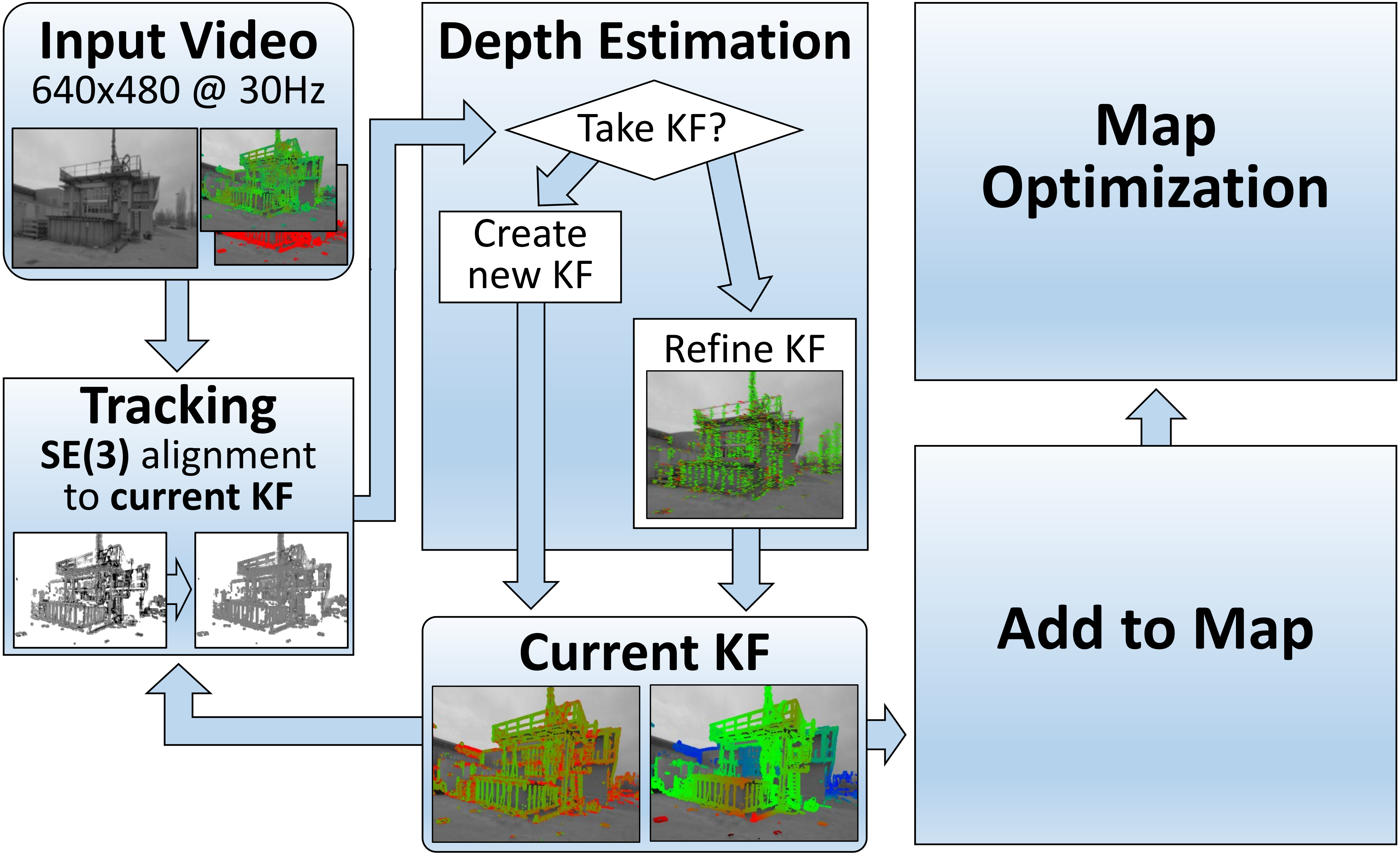
Refine KF

Current KF

Map Optimization

Add to Map

Overview



Input Video

640x480 @ 30Hz

Tracking

SE(3) alignment to **current KF**

Depth Estimation

Take KF?

Create new KF

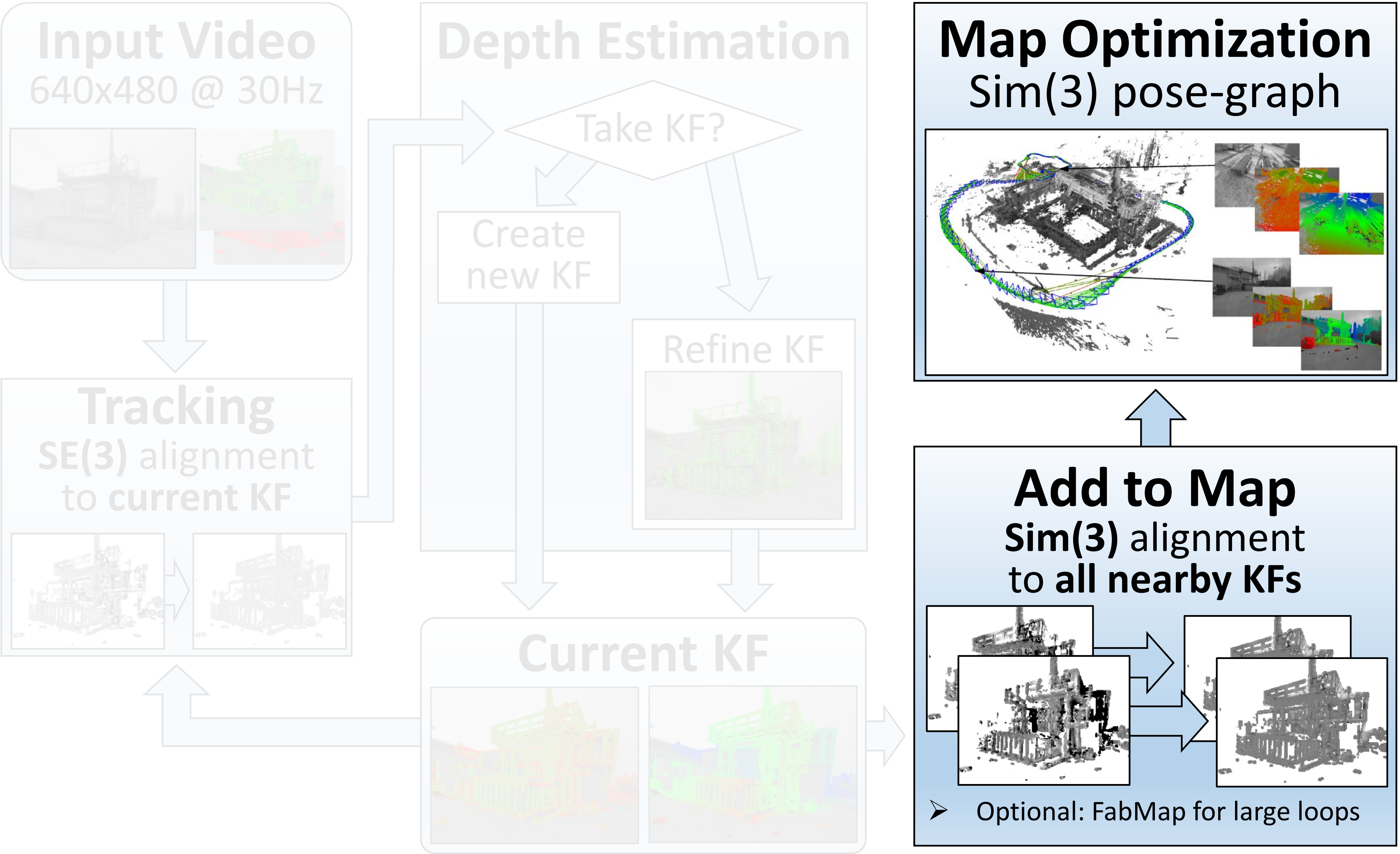
Refine KF

Current KF

Map Optimization

Add to Map

Overview



Input Video

640x480 @ 30Hz

Tracking

SE(3) alignment

to **current KF**

Depth Estimation

Take KF?

Create new KF

Refine KF

Current KF

Map Optimization

Sim(3) pose-graph

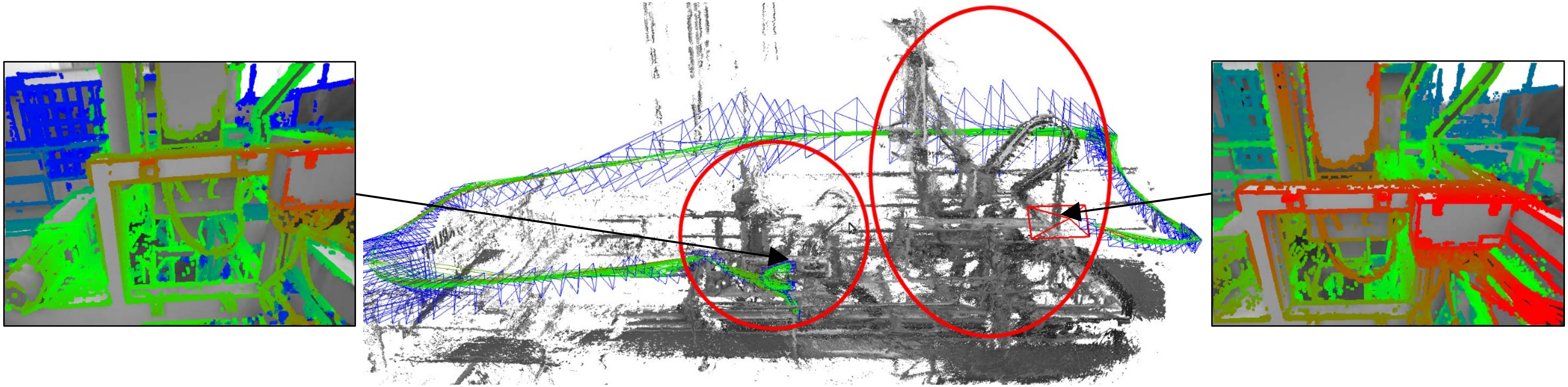
Add to Map

Sim(3) alignment

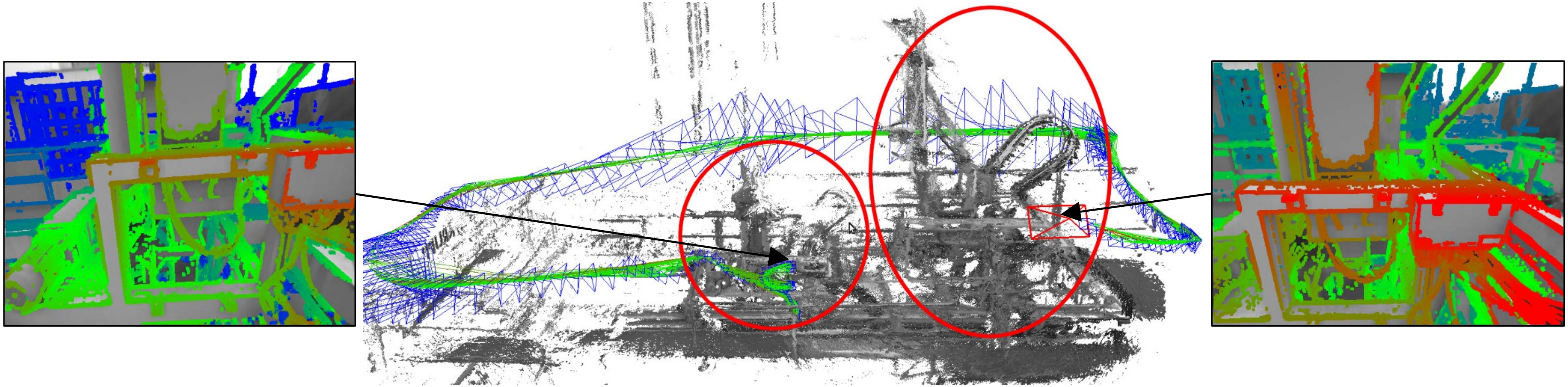
to **all nearby KFs**

- Optional: FabMap for large loops

Global Mapping



Global Mapping

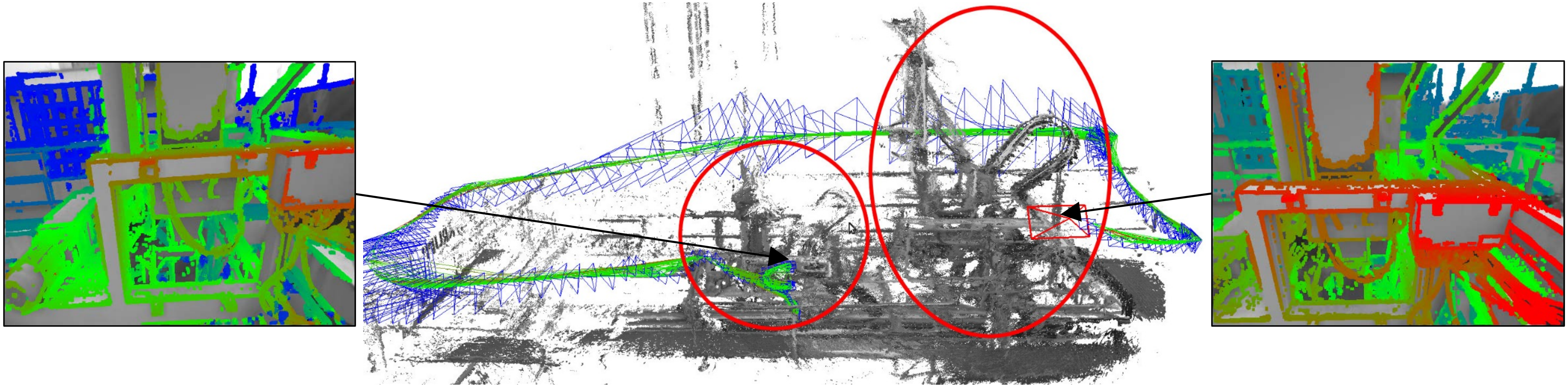


- **Direct Tracking with scale (on Sim(3)):**

$$E(\underset{\mathfrak{se}(3)}{\underset{\nearrow}{\boldsymbol{\xi}}}) = \sum_{\mathbf{x}\in\Omega_1} \left(I_1(\mathbf{x}) - I_2(\mathbf{x}')\right)^2$$

with $\mathbf{x}' := \omega(\mathbf{x}, D_1(\mathbf{x}), \boldsymbol{\xi})$ (warped point)

Global Mapping



➢ Direct Tracking with scale (on Sim(3)):

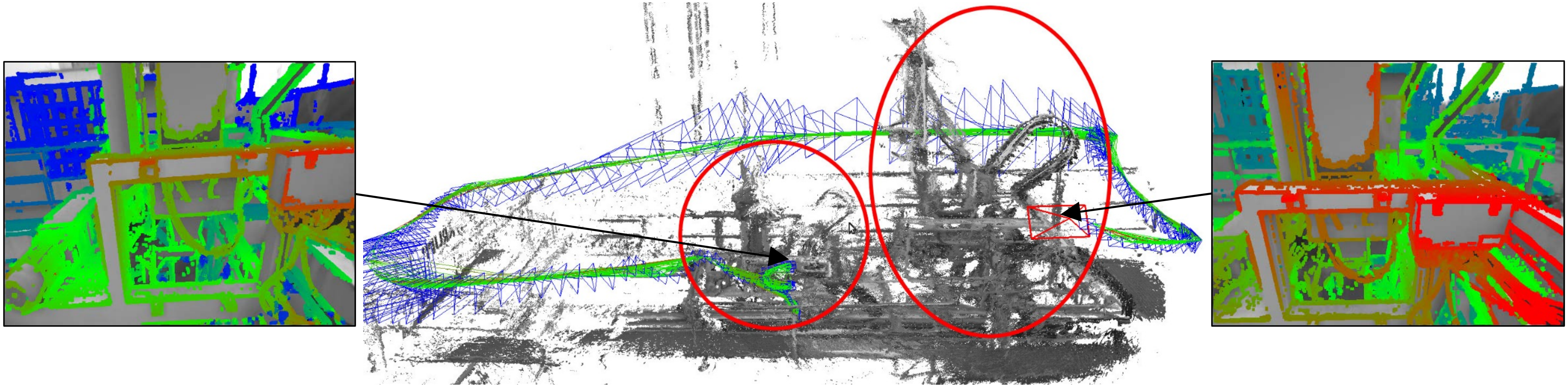
$$E(\boldsymbol{\xi}) = \sum_{\mathbf{x}\in\Omega_1}\Big(\big(I_1(\mathbf{x}) - I_2(\mathbf{x}')\big)^2 + \big([\mathbf{x}']_3 - D_2(\mathbf{x}')\big)^2\Big)$$

~~$\mathfrak{se}(3)$~~

$\mathfrak{sim}(3)$ with $\mathbf{x}' := \omega(\mathbf{x}, D_1(\mathbf{x}), \boldsymbol{\xi})$ (warped point)

▪

Global Mapping



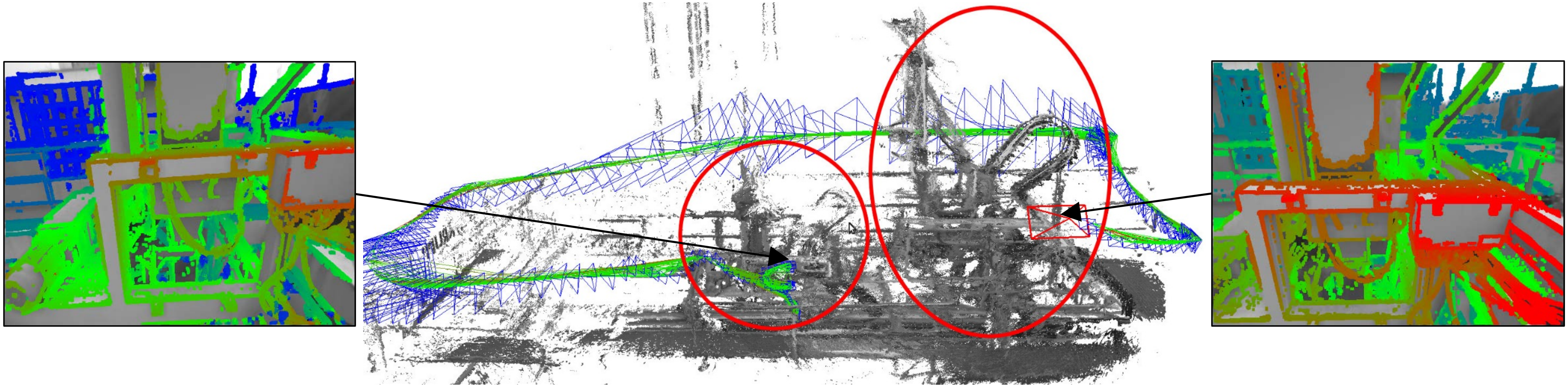
- **Direct Tracking with scale (on Sim(3)):**

$$E(\underset{\overset{\nearrow}{\cancel{\mathfrak{se}(3)}}}{\boldsymbol{\xi}}) = \sum_{\mathbf{x}\in\Omega_1}\Big(\big(I_1(\mathbf{x}) - I_2(\mathbf{x}')\big)^2 + \big([\mathbf{x}']_3 - D_2(\mathbf{x}')\big)^2\Big)$$

$\mathfrak{sim}(3)$ with $\mathbf{x}' := \omega(\mathbf{x}, D_1(\mathbf{x}), \boldsymbol{\xi})$ (warped point)

+ GN optimization + multi-resolution + Huber norm + statistical norm.

Global Mapping



- **Direct Tracking with scale (on Sim(3)):**

$$E(\boldsymbol{\xi}) = \sum_{\mathbf{x}\in\Omega_1} \Big(\big(I_1(\mathbf{x}) - I_2(\mathbf{x}')\big)^2 + \big([\mathbf{x}']_3 - D_2(\mathbf{x}')\big)^2 \Big)$$

~~$\mathfrak{se}(3)$~~

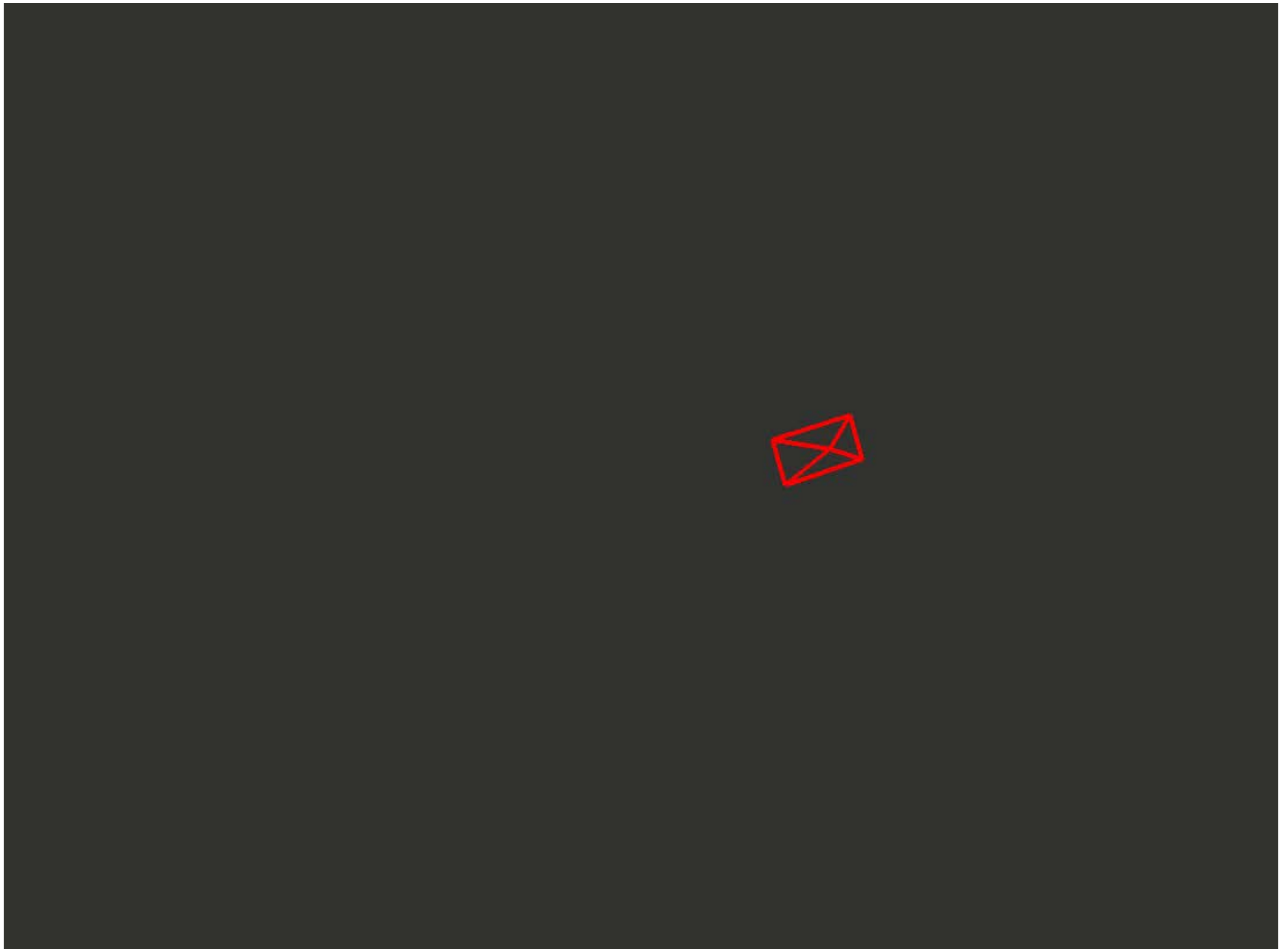
$\mathfrak{sim}(3)$ with $\mathbf{x}' := \omega(\mathbf{x}, D_1(\mathbf{x}), \boldsymbol{\xi})$ (warped point)

+ GN optimization + multi-resolution + Huber norm + statistical norm.

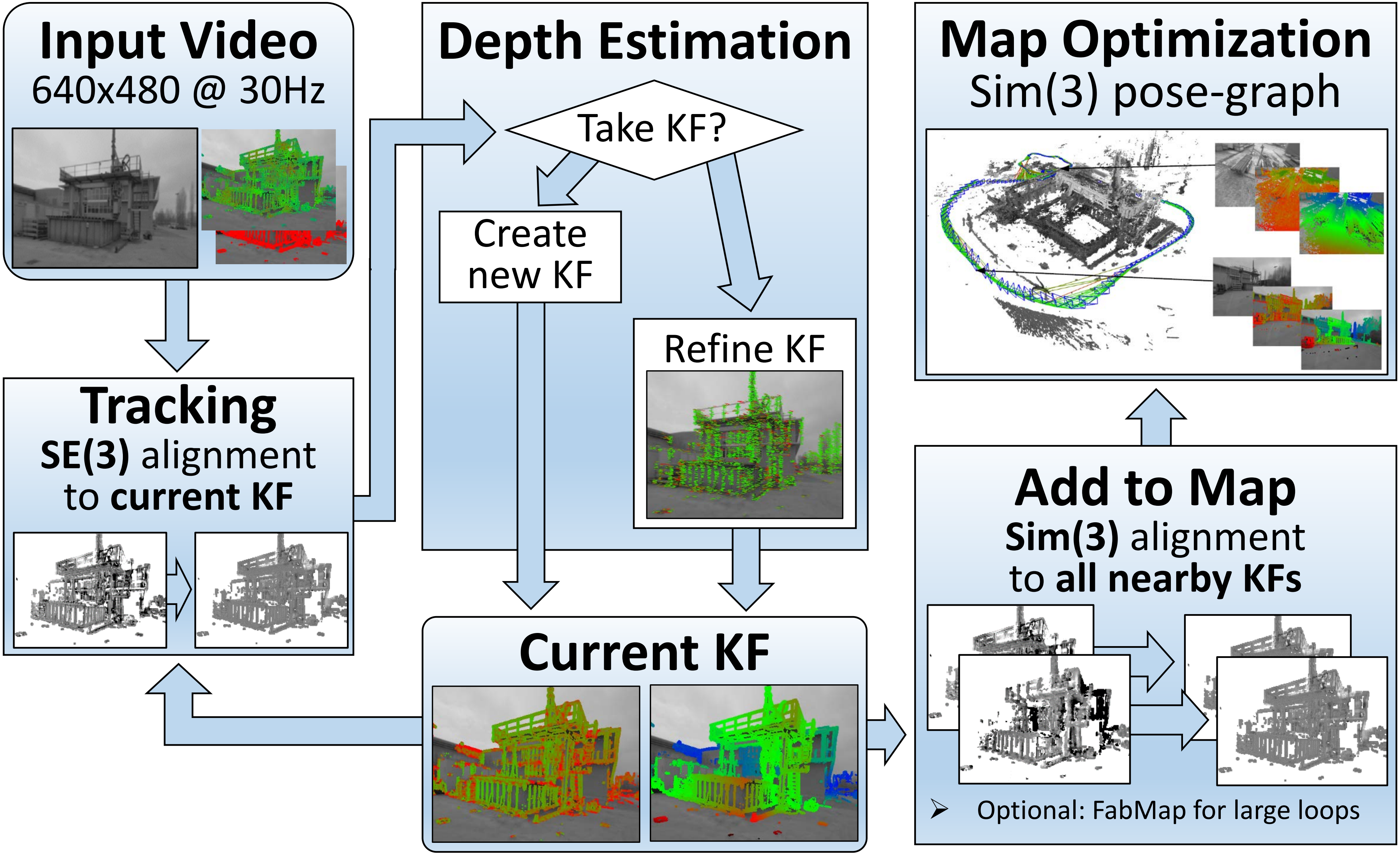
- **Optimize pose-graph on Sim(3)**

$$E(\boldsymbol{\xi}_{1W} \dots \boldsymbol{\xi}_{nW}) := \sum_{(\boldsymbol{\xi}_{ij}, \boldsymbol{\Sigma}_{ij})\in\mathcal{E}} (\boldsymbol{\xi}_{ij} \circ \boldsymbol{\xi}_{iW}^{-1} \circ \boldsymbol{\xi}_{jW})^T \boldsymbol{\Sigma}_{ij}^{-1} (\boldsymbol{\xi}_{ij} \circ \boldsymbol{\xi}_{iW}^{-1} \circ \boldsymbol{\xi}_{jW}).$$

Global Mapping



Overview



Input Video

640x480 @ 30Hz

Tracking

SE(3) alignment
to **current KF**

Depth Estimation

Take KF?

Create
new KF

Refine KF

Current KF

Map Optimization

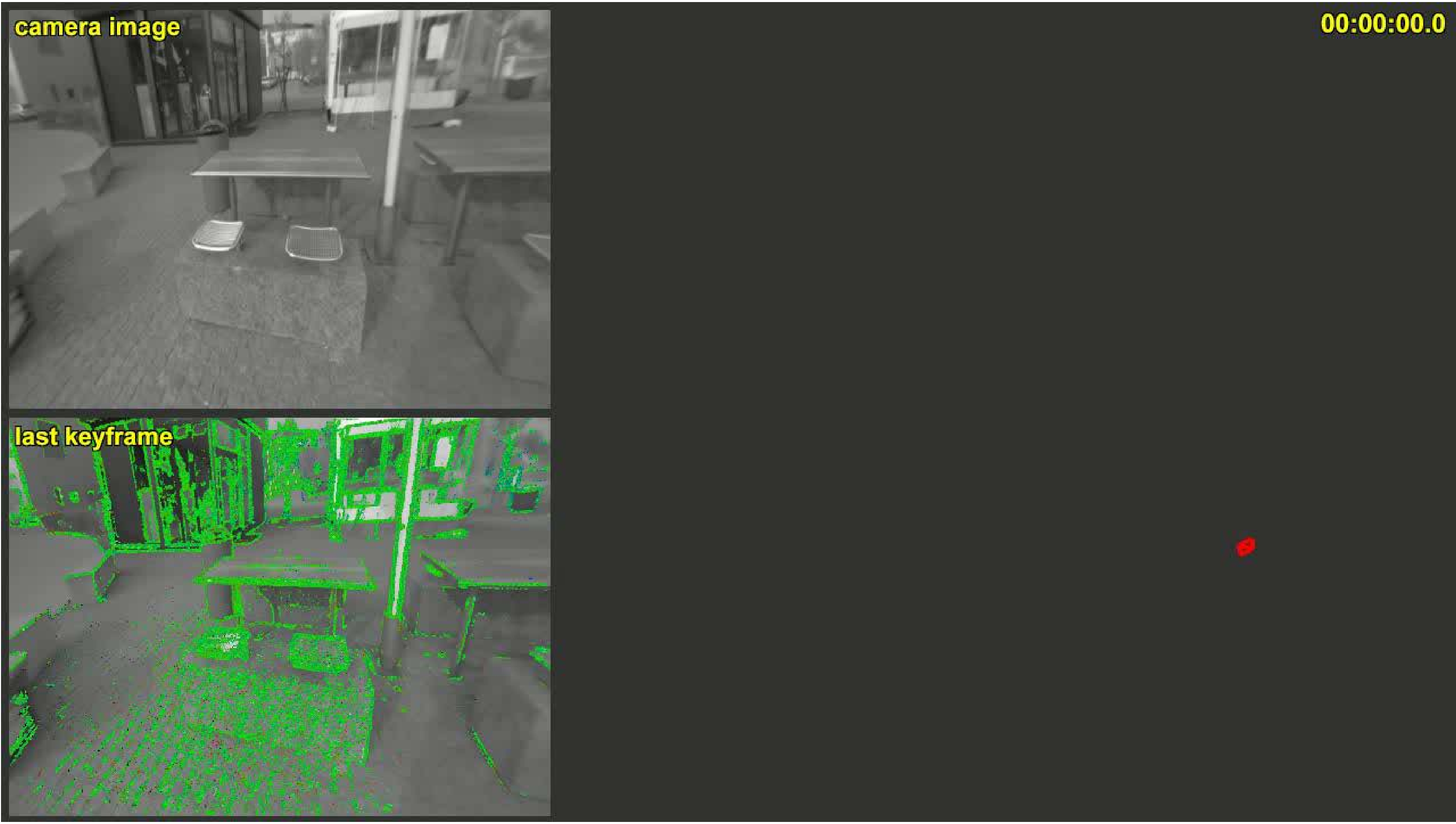
Sim(3) pose-graph

Add to Map

Sim(3) alignment
to **all nearby KFs**

- Optional: FabMap for large loops

Results



6 minutes, 640x480@50fps:

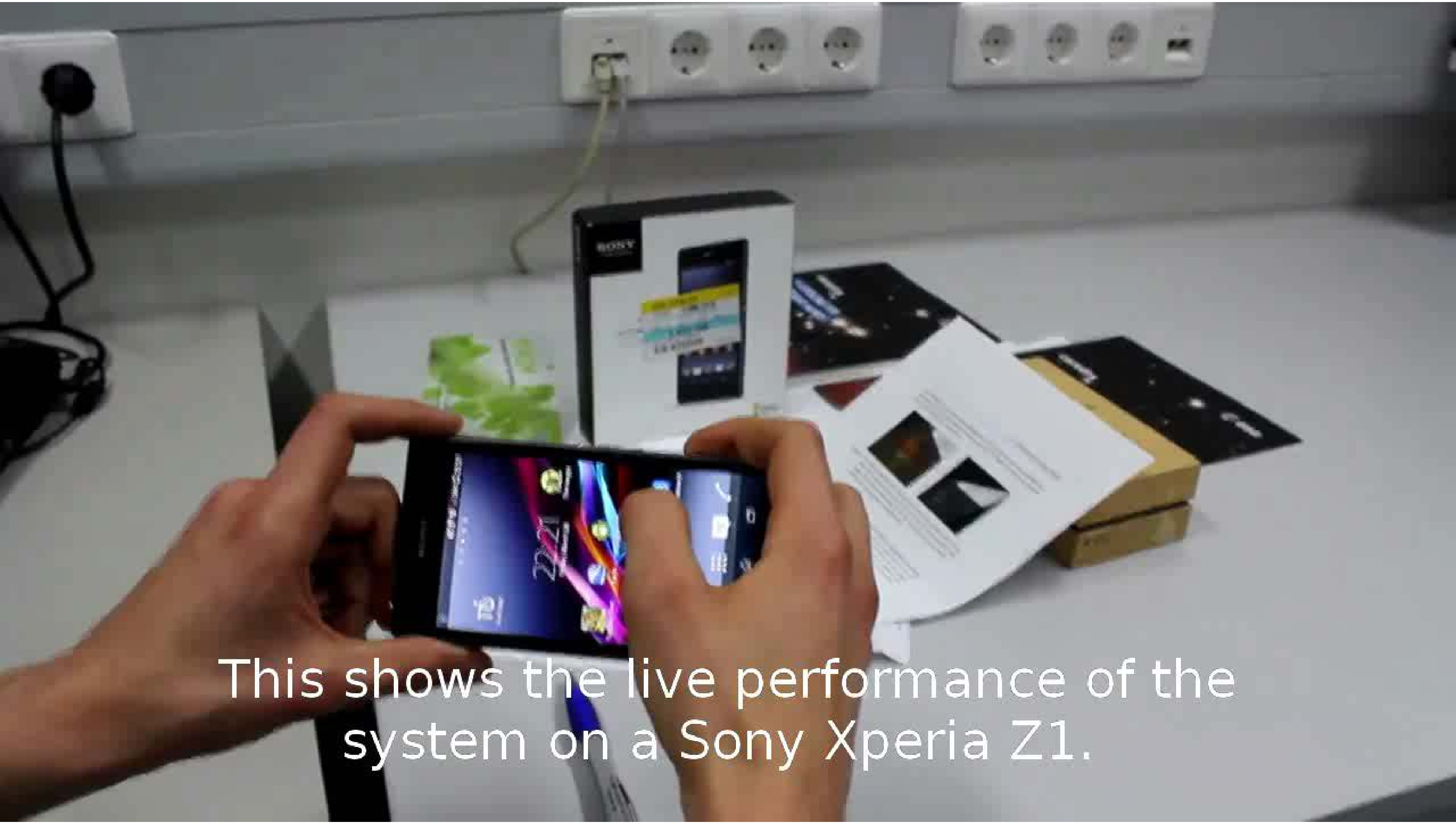
16.000 Tracked Frames, 800 Keyframes; 11.000 Constraints; 51 Million Points

Results

12 minutes, 640x480@50fps:

36.000 Tracked Frames, 1.000 Keyframes; 18.000 Constraints; 100 Million Points

Results

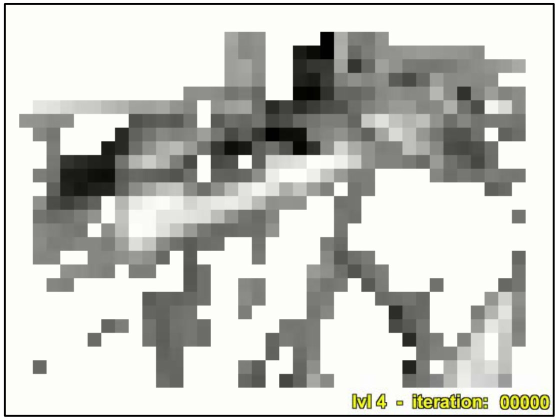

This shows the live performance of the system on a Sony Xperia Z1.

Semi-Dense Visual Odometry for AR on a Smartphone; *T. Schöps, J. Engel, D. Cremers*; ISMAR ´14.

Key Ingredients

- Direct Tracking

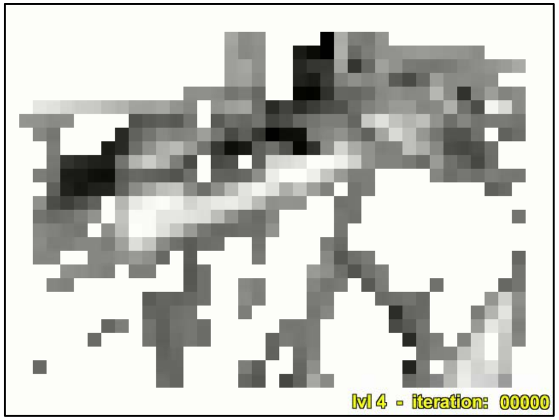
$$E(\boldsymbol{\xi}) = \sum_i w_i(\boldsymbol{\xi}) \left(I_{\mathrm{ref}}(\mathbf{p}_i) - I(\omega(\mathbf{p}_i, D_{\mathrm{ref}}(\mathbf{p}_i), \boldsymbol{\xi}))\right)^2$$



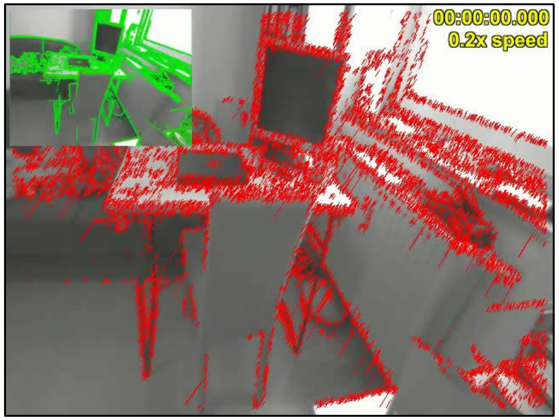
Key Ingredients

- Direct Tracking

$$E(\boldsymbol{\xi}) = \sum_i w_i(\boldsymbol{\xi}) \left(I_{\text{ref}}(\mathbf{p}_i) - I(\omega(\mathbf{p}_i, D_{\text{ref}}(\mathbf{p}_i), \boldsymbol{\xi}))\right)^2$$

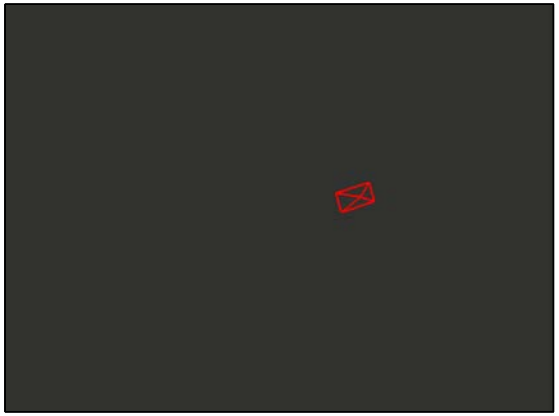


- Semi-Dense Stereo
  - filter over many small-baseline frames
  - strict information selection



- Pose-Graph on Sim(3)

$$E(\boldsymbol{\xi}_{1W} \dots \boldsymbol{\xi}_{nW}) := \sum_{(\boldsymbol{\xi}_{ij}, \boldsymbol{\Sigma}_{ij}) \in \mathcal{E}} (\boldsymbol{\xi}_{ij} \circ \boldsymbol{\xi}_{iW}^{-1} \circ \boldsymbol{\xi}_{jW})^T \boldsymbol{\Sigma}_{ij}^{-1} (\boldsymbol{\xi}_{ij} \circ \boldsymbol{\xi}_{iW}^{-1} \circ \boldsymbol{\xi}_{jW}).$$



LSD-SLAM

- **Large-scale** direct mono-SLAM
- **Fully direct** (no keypoints / features)
- **Real-time** even on CPU
- **Open-source** code & data-sets