Autonomous Navigation for Flying Robots

Lecture 8.3: Direct Methods for Visual SLAM

Jürgen Sturm

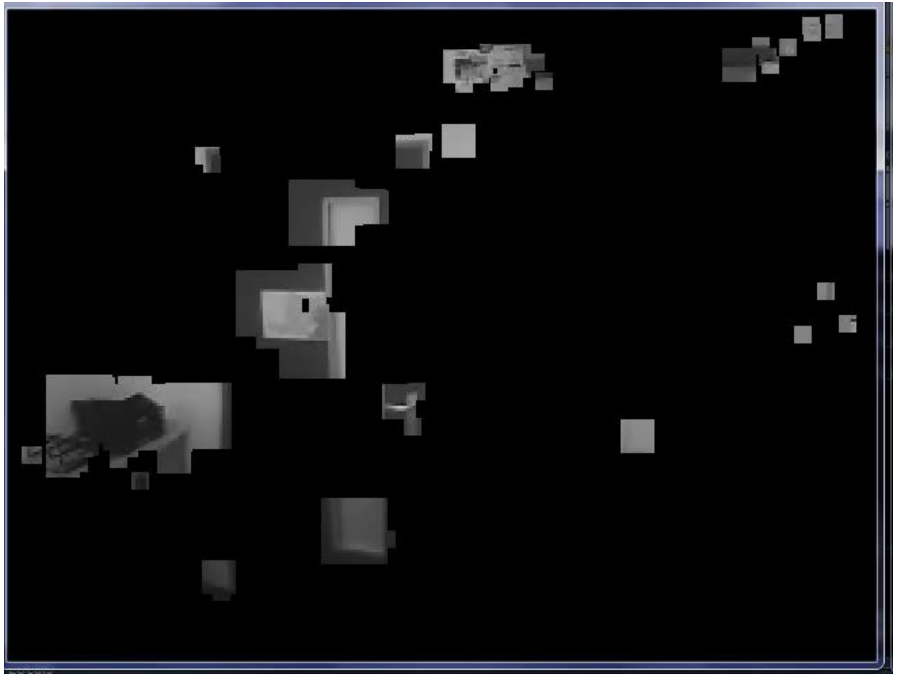
Technische Universität München

Feature-Based Visual SLAM

- Video feed from quadrocopter

Feature-Based Visual SLAM

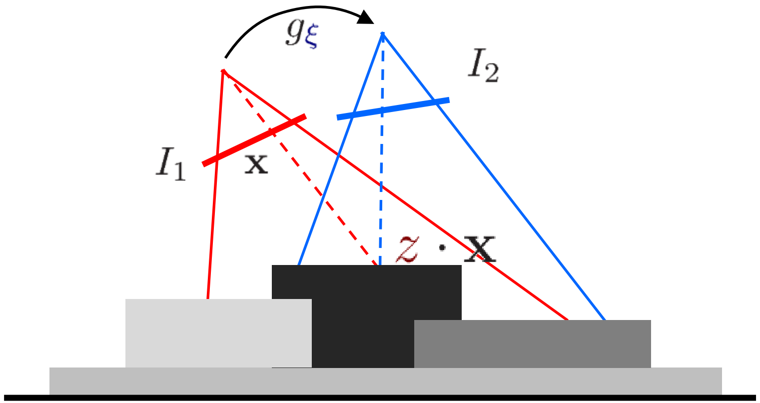
- What PTAM actually sees



Direct Visual Odometry

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

- How can we use more/all information from the image?
- Idea



- Photo-consistency constraint

$$I_1(\mathbf{x}) = I_2\left(\pi(g_\xi(z \cdot \mathbf{x}))\right)$$ holds for all pixels $\mathbf{x}$

How to deal with noise?

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

- Photo-consistency constraint will not perfectly hold
  - Sensor noise
  - Pose error
  - Reflections, specular surfaces
  - Dynamic objects (e.g., walking people)
- Residuals will be non-zero

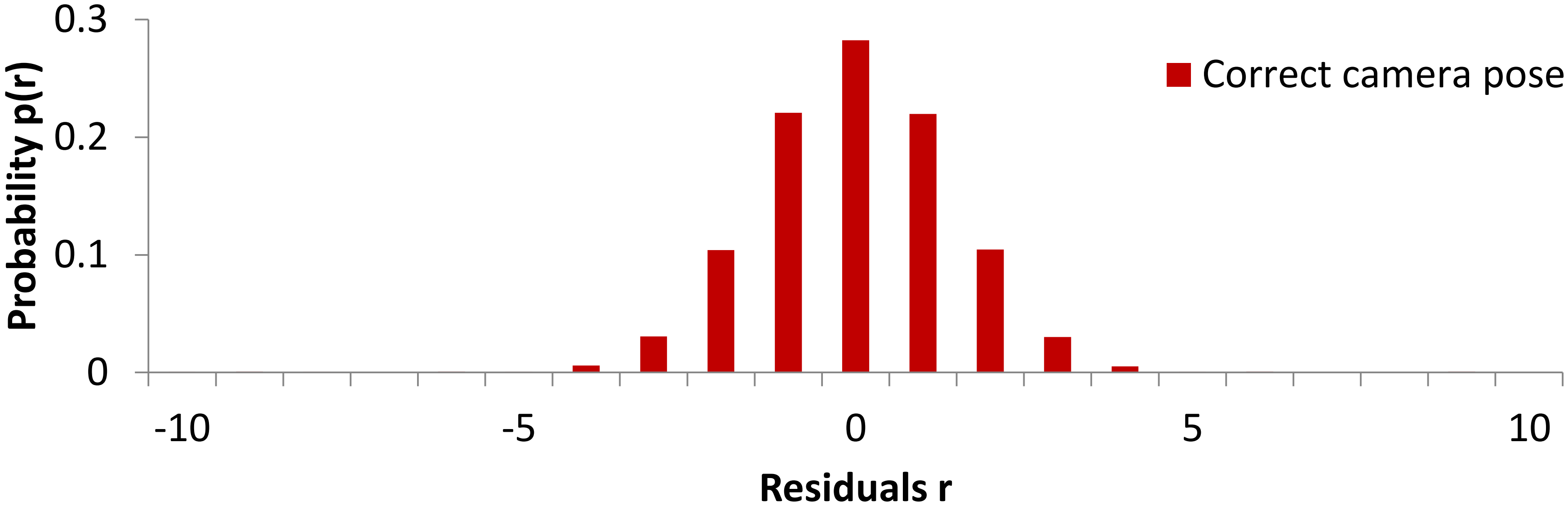
$$r = I_1(\mathbf{x}) - I_2\left(\pi(g_\xi(z \cdot \mathbf{x}))\right)$$

- Residual distribution $p(r)$

Residual Distribution

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

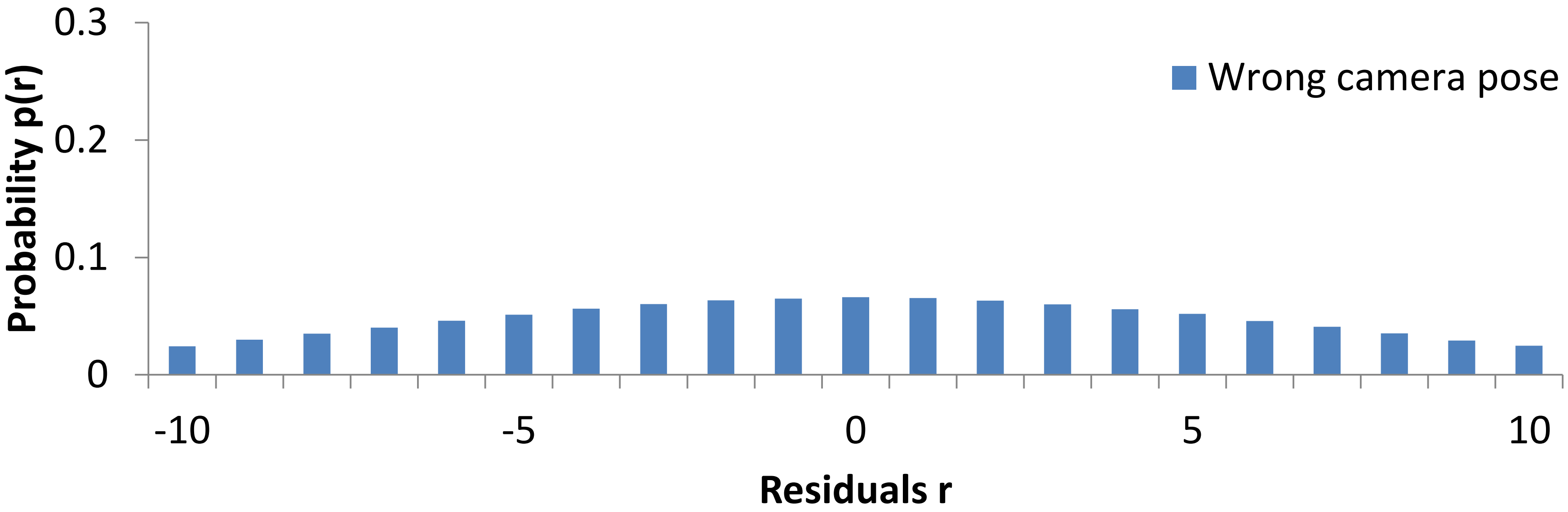
- Zero-mean, peaked distribution
- Example: Correct camera pose



Residual Distribution

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

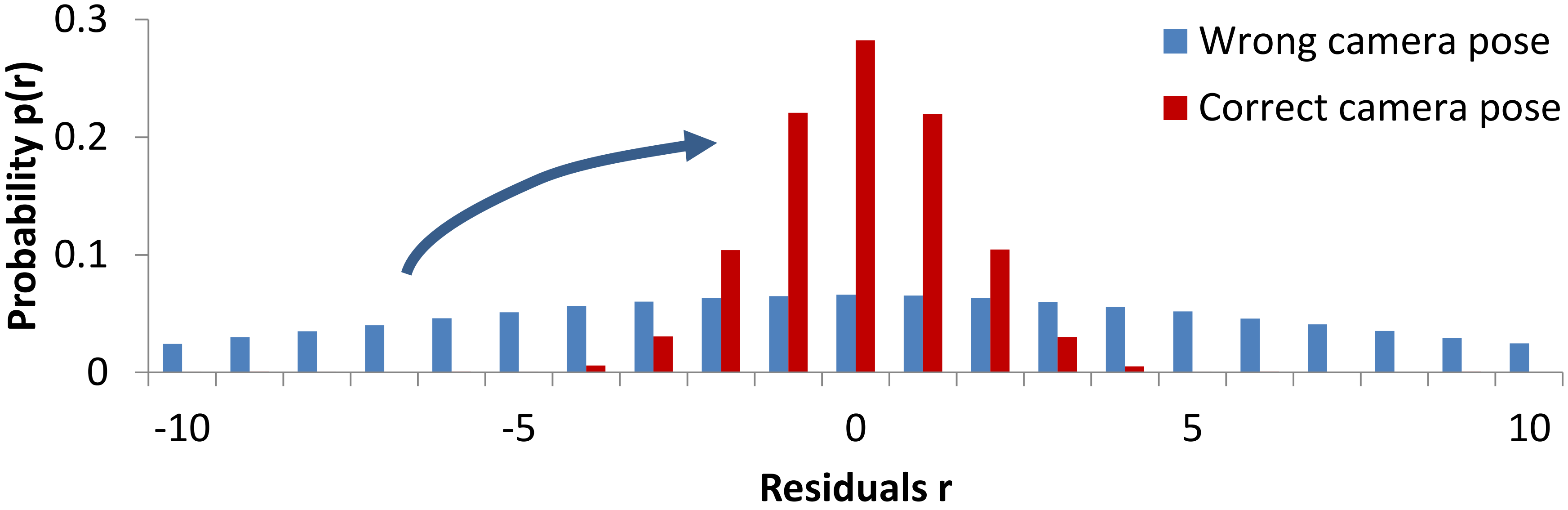
- Zero-mean, peaked distribution
- Example: Correct camera pose



Residual Distribution

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

- **Goal:** Find the camera pose that maximizes the observation likelihood



Motion Estimation

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

- **Goal:** Find the camera pose that maximizes the observation likelihood

$$\boldsymbol{\xi}^* = \arg\max_{\boldsymbol{\xi}} \prod_i p(r_i(\boldsymbol{\xi}))$$

compute over all pixels

- Assume pixel-wise residuals are conditionally independent
- How can we solve this optimization problem?

Approach
[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

- Take negative logarithm

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \sum_i - \log p(r_i(\boldsymbol{\xi}))$$

- Set derivative to zero

$$\sum_i \frac{\partial \log p(r_i(\boldsymbol{\xi}))}{\partial \boldsymbol{\xi}} = \sum_i \frac{\partial \log p(r_i)}{\partial r_i} \frac{\partial r_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \stackrel{!}{=} 0$$

Robust Cost Functions

[Kerl, Sturm, Cremers; ICRA 2013]

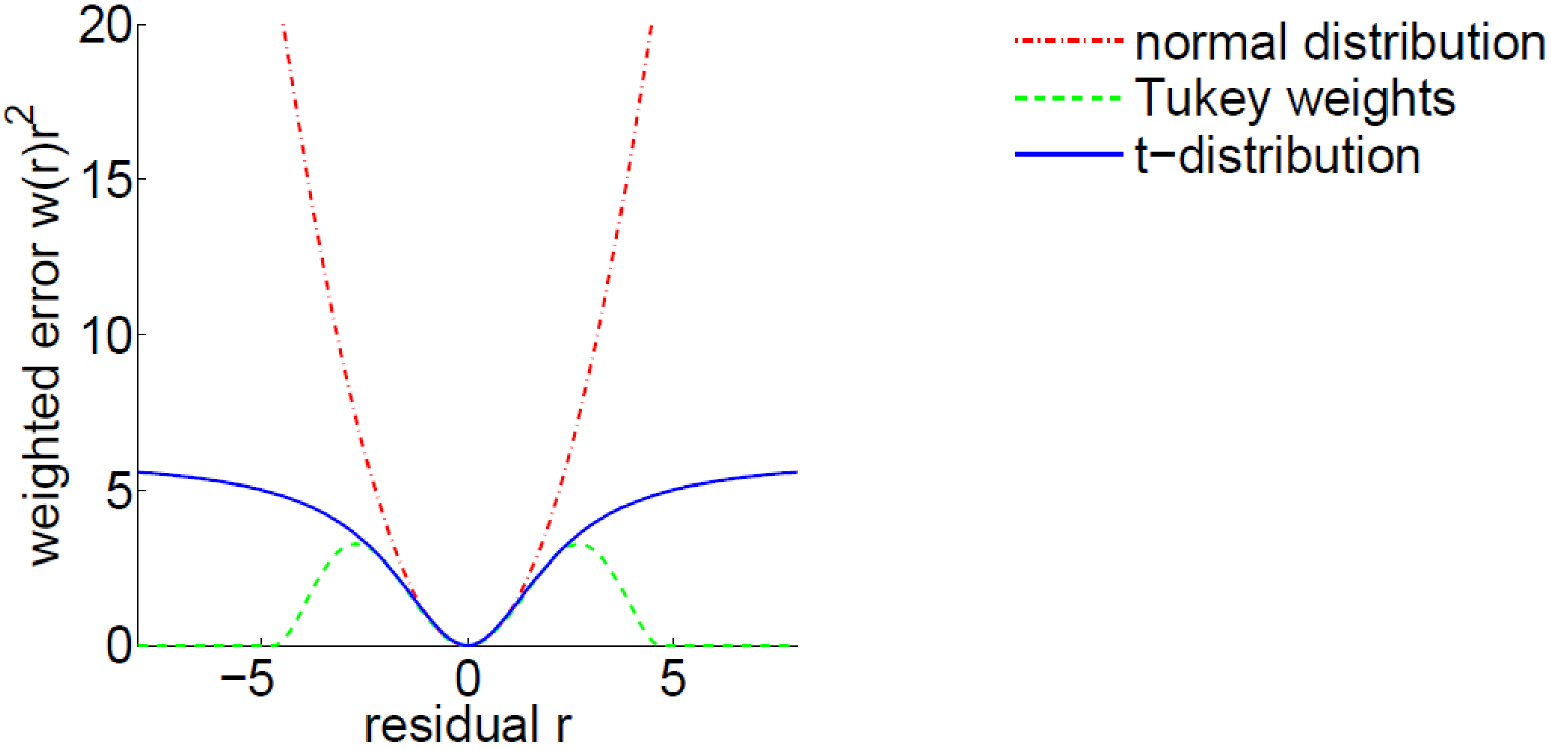
- Quadratic cost term is not robust
- Rewrite as a weighted least squares problem

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \sum_i w(r_i)(r_i(\boldsymbol{\xi}))^2$$

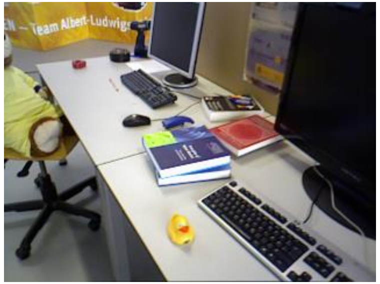
with weights $w(r_i) = \frac{\partial \log p(r_i)}{\partial r_i} \frac{1}{r_i}$

- $r_i(\boldsymbol{\xi})$ is non-linear in $\boldsymbol{\xi}$
- Need to linearize, solve, and iterate (Gauss-Newton method)

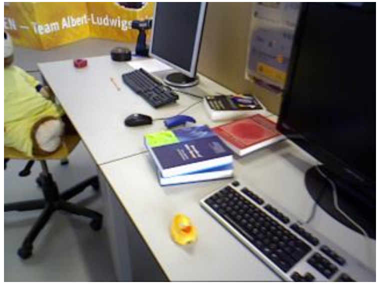
Weighted Error
[Kerl, Sturm, Cremers; ICRA 2013]



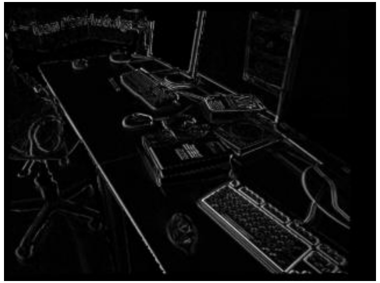
Example
[Kerl, Sturm, Cremers; ICRA 2013]



First input image



Second input image



Residuals

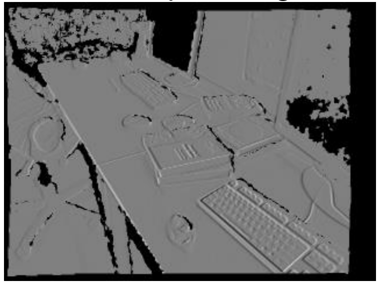
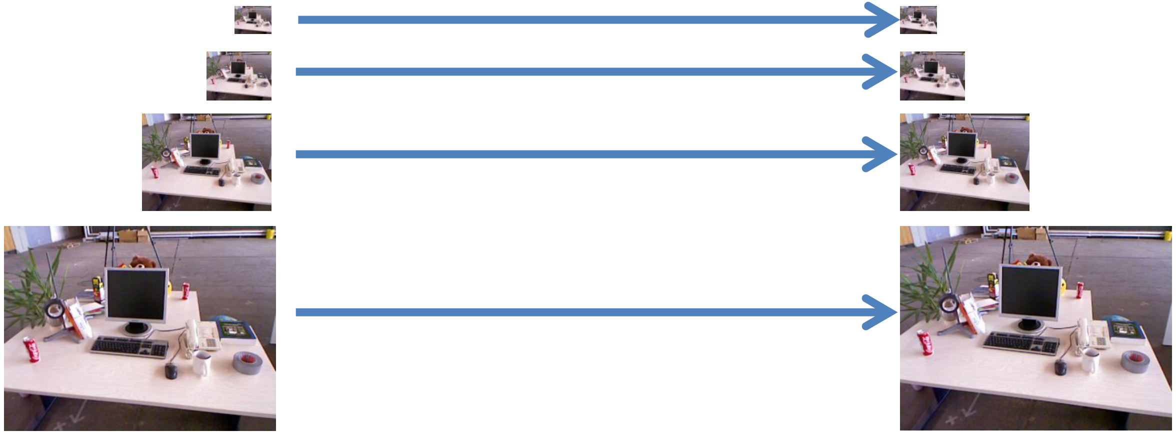


Image Jacobian for
camera motion along x axis

Coarse-to-Fine

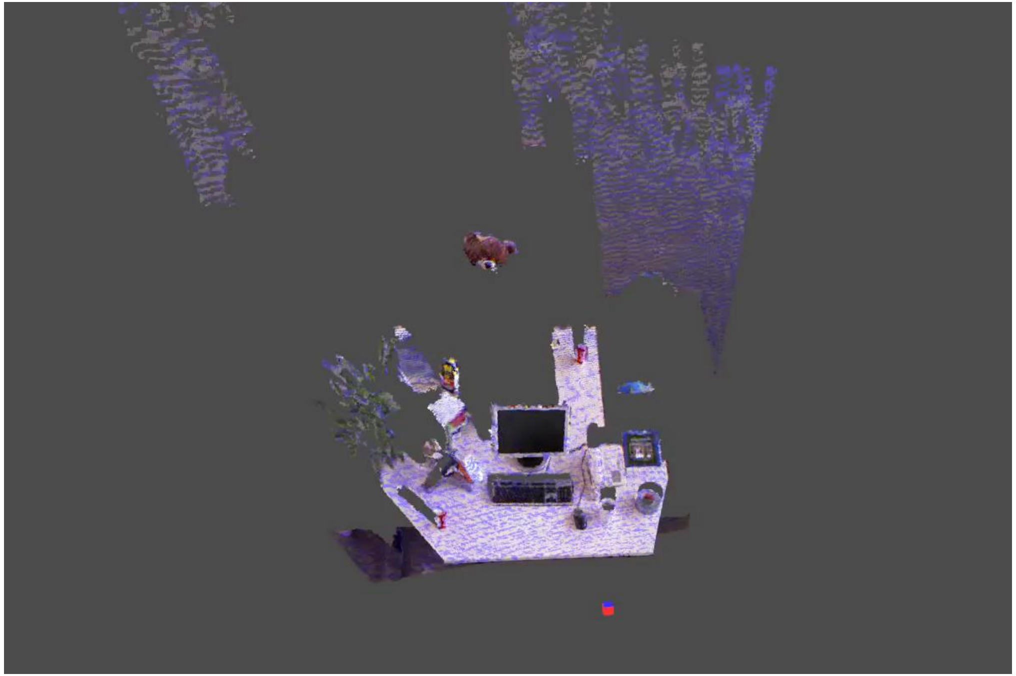
[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]

- Linearization only holds for small motions
- Coarse-to-fine scheme
- Image pyramids



Direct Visual Odometry: Results

[Steinbrücker, Sturm, Cremers; ICCV LDRMC 2011]



Real-Time Visual Odometry from Dense RGB-D Images (F. Steinbruecker, J. Sturm, D. Cremers), In Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV), 2011.

Direct SLAM

[Kerl, Sturm, Cremers; IROS 2013]

- Dense Visual Odometry
  - Input: Two RGB-D frames
  - Output: Relative pose
  - Runs in real-time on single CPU core
- Use this in pose graph SLAM
  - Select keyframes
  - Detect loop-closures
  - Build and optimize pose graph (using g2o)

Direct SLAM
[Kerl, Sturm, Cremers; IROS 2013]

Dense Visual SLAM for RGB-D Cameras (C. Kerl, J. Sturm, D. Cremers), In Proc. of the Int. Conf. on Intelligent Robot Systems (IROS), 2013. http://youtu.be/jNbYcw_dmcQ

Large-Scale 3D Reconstruction

[Steinbrücker, Kerl, Sturm, Cremers; ICCV 2013]

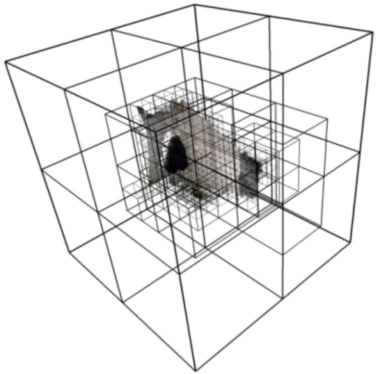
- **We have:** Optimized pose graph
- **We want:** High-resolution 3D map

- **Problem:** High-resolution voxel grids consume much memory (grows cubically)
  - 512^3 voxels, 24 byte per voxel → 3.2 GB
  - 1024^3 voxels, 24 byte per voxel → 24 GB
  - …

Large-Scale 3D Reconstruction

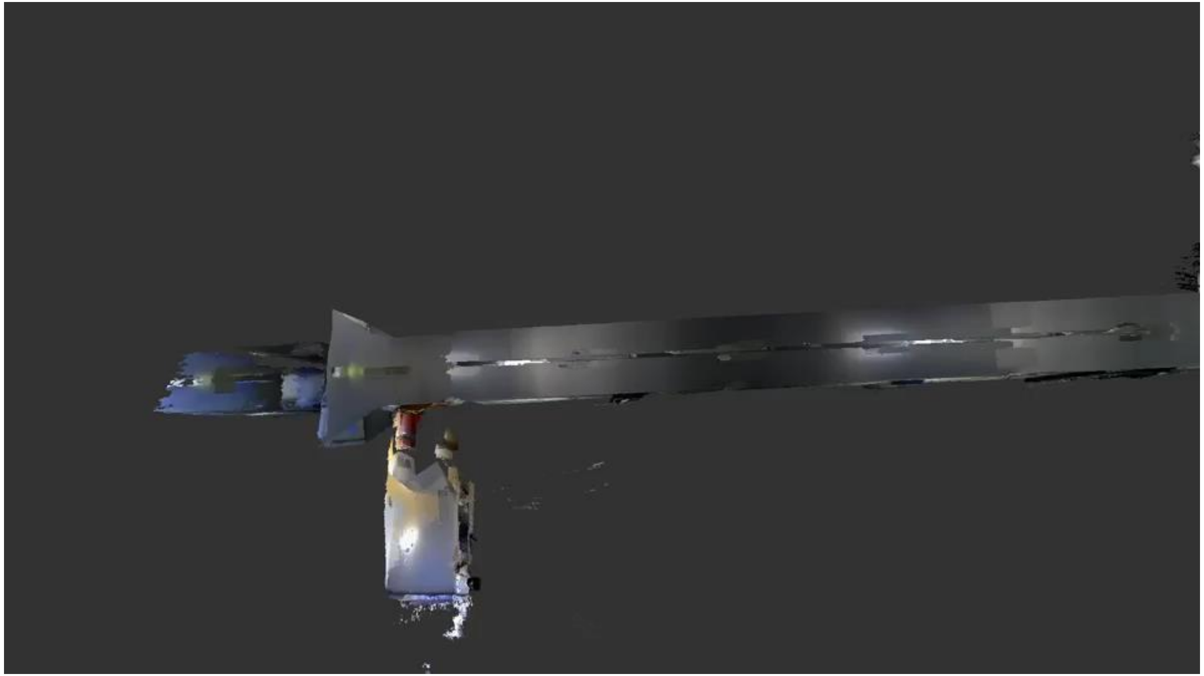
[Steinbrücker, Kerl, Sturm, Cremers; ICCV 2013]

- Save data in oct-tree data structure
- Only allocate cells that are close to the surface
- Store geometry at multiple resolutions
- Tree can grow dynamically (no fixed size)



Large-Scale 3D Reconstruction

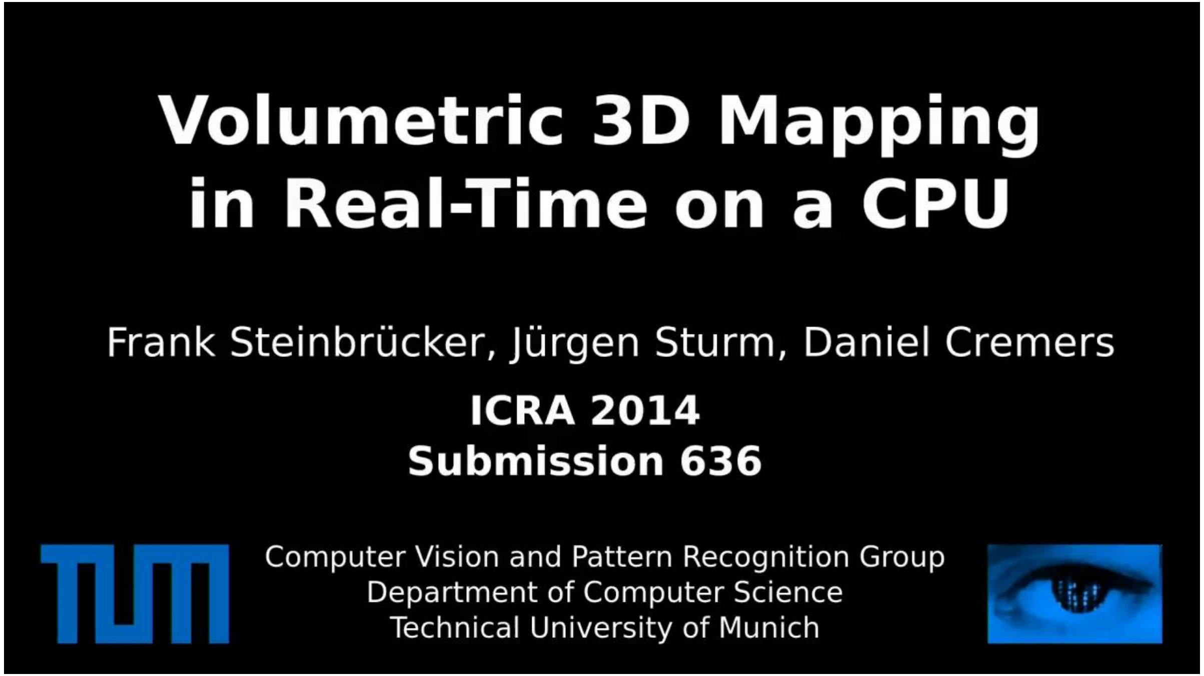
[Steinbrücker, Kerl, Sturm, Cremers; ICCV 2013]



Large-Scale Multi-Resolution Surface Reconstruction from RGB-D Sequences (F. Steinbruecker, C. Kerl, J. Sturm, D. Cremers), In IEEE International Conference on Computer Vision (ICCV), 2013. http://youtu.be/J37f9vH-CPc

3D Mapping in Real-Time on a CPU

[Steinbrücker, Sturm, Cremers; ICRA 2014]



Volumetric 3D Mapping in Real-Time on a CPU (F. Steinbruecker, J. Sturm, D. Cremers), In Int. Conf. on Robotics and Automation, 2014.
http://youtu.be/7s9JePSln-M

Direct SLAM with a Monocular Camera

[Engel, Sturm, Cremers; ICCV 2013]

Semi-Dense Visual Odometry for a Monocular Camera (J. Engel, J. Sturm, D. Cremers), In IEEE International Conference on Computer Vision (ICCV), 2013.
http://youtu.be/LZChzEcLNzI

Lessons Learned

- Direct visual odometry
- Photoconsistency constraint
- Loop closing
- Large-scale 3D reconstruction
- Software available as open-source
  https://github.com/tum-vision