Technische Universität München

Efficient Online Surface Correction for Real-time Large-Scale 3D Reconstruction

Robert Maier* Raphael Schaller* Daniel Cremers

Computer Vision Group

* equal contribution

Motivation

Real-time large-scale 3D reconstruction:

- Global **pose optimization** to reduce drift
- Usually **no surface update** on pose changes

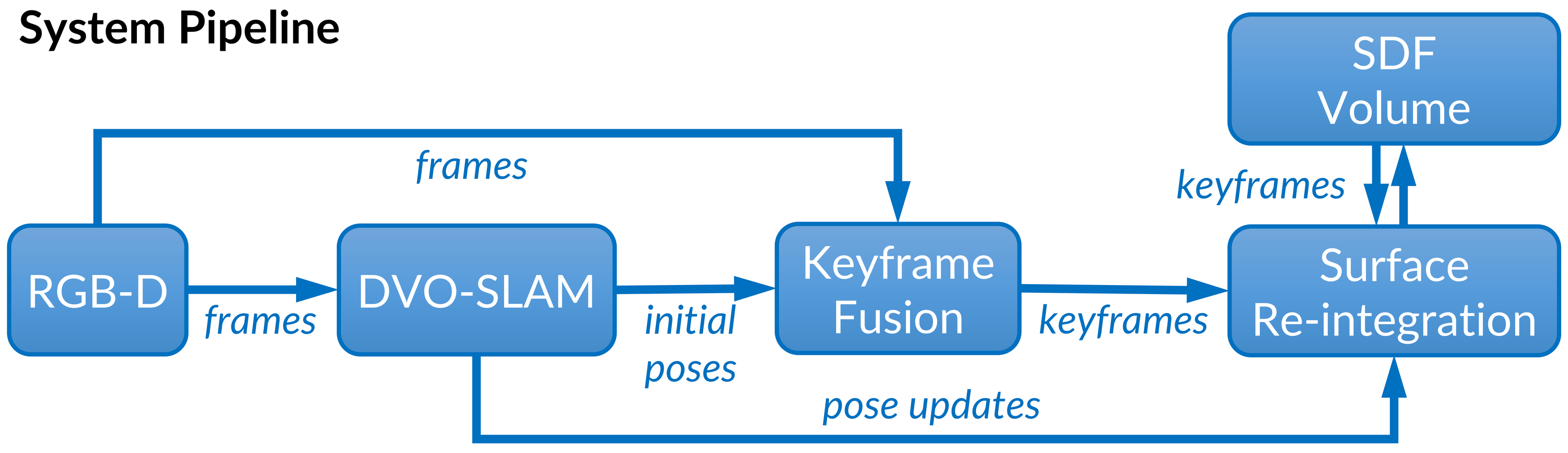
Goal: correct 3D surface on-the-fly on pose changes

Contributions

- **Efficient on-the-fly surface correction** for large-scale dense 3D reconstruction (single GPU)
  - **Keyframe fusion** of RGB-D frames using different keyframe strategies
  - **Re-integrate fused keyframes** into sparse SDF volume **on pose updates**
  - **Efficient re-integration strategy:** reduced host-GPU-streaming
  - **93% more efficient** than state-of-the-art (equivalent surface quality)
- Combination with dense Visual SLAM system (CPU)

3D Reconstruction System

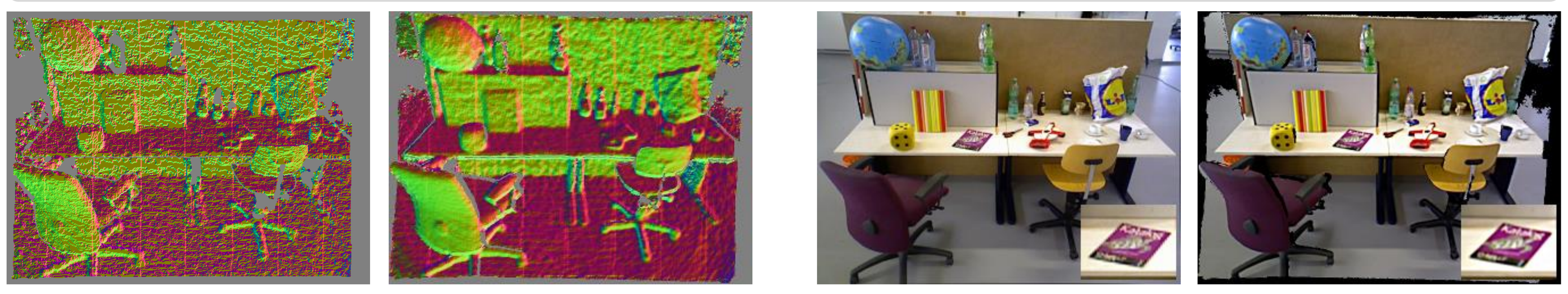
System Pipeline



DVO-SLAM: Dense Visual RGB-D SLAM [1]

- **Frame-to-(key)frame tracking** using robust dense visual odometry
- **Loop closure detection** and continuous **pose graph optimization**

Keyframe Fusion



Depth fusion
(weighted mean of warped input depth maps)

Color fusion
(unsharp masking, blurriness, weighted median)

Efficient Online Surface Re-Integration

Keyframe Strategies

- Selection of **independent keyframes for fusion**
  → number of fused frames per keyframe *k*
- Strategies: Constant, DVO, Distance, Overlap

On-the-fly Surface Correction

- Integrate keyframes into SDF volume with original poses
- On DVO-SLAM pose graph updates:
  - **Select *m changed* keyframes** for re-integration
  - **De-integrate** keyframes at original poses
  - **Re-integrate** on-the-fly with updated poses

Re-integration Strategy

- BundleFusion [2] strategy: select *m most-moved frames*
- Better: **select group of *most-moved m consecutive keyframes***

$$j^* = \underset{j \in [1, K-m+1]}{\operatorname{argmax}} \sum_{i=j}^{j+m-1} \| st_i - st'_i \|$$

→ **Significantly reduced GPU-host-streaming!**

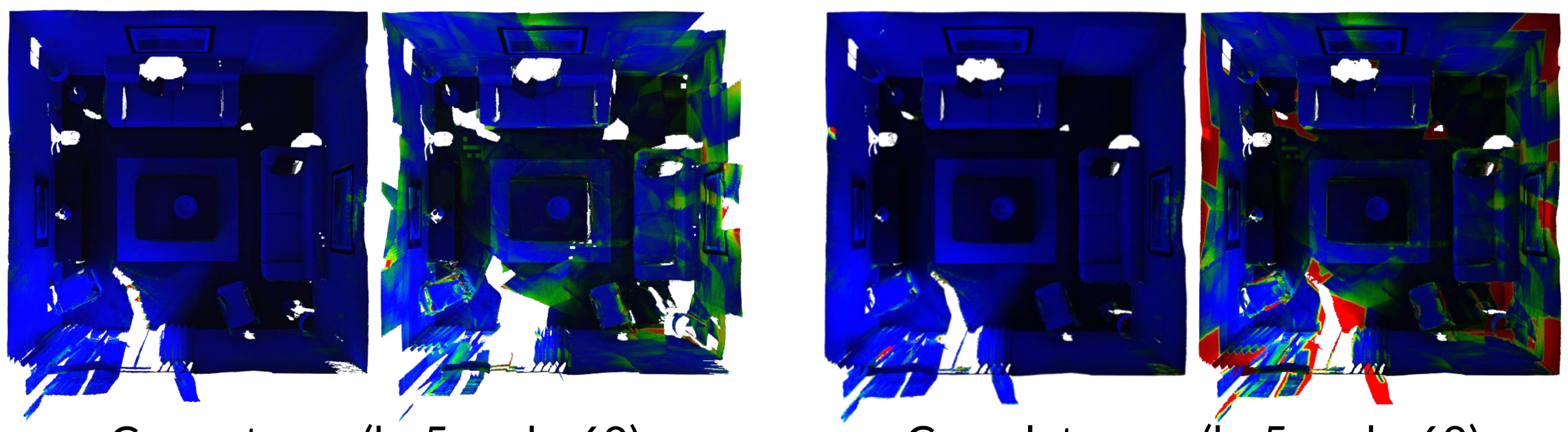
- Example: BundleFusion (12, 8, 13, 5, 3) vs. ours (4 to 8)

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Distance	1	3	4	3	5	4	1	7	2	1	1	8	6	2	0
Distance (sum)			16	19	17	**20**	19	15	12	19	18	18	17		

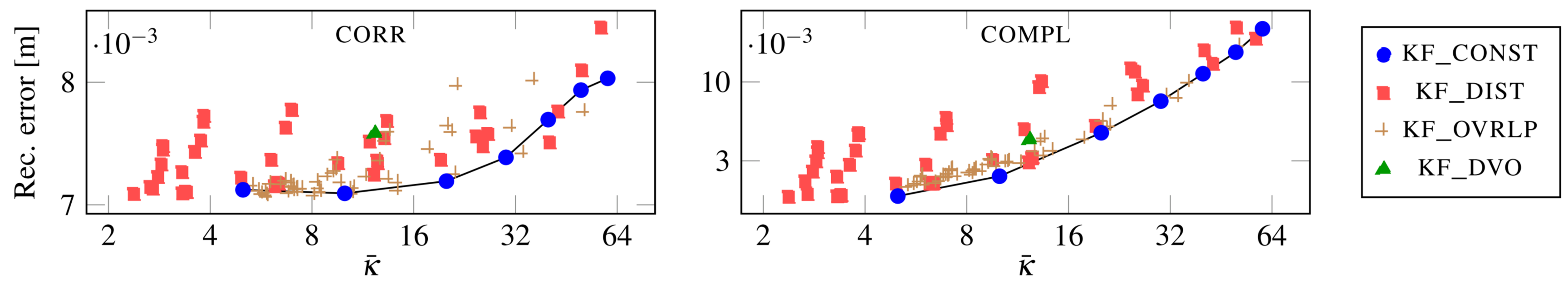
Evaluation and Experimental Results

Surface Completeness/Correctness vs. Frames per Keyframe

AUG_ICL/Liv1

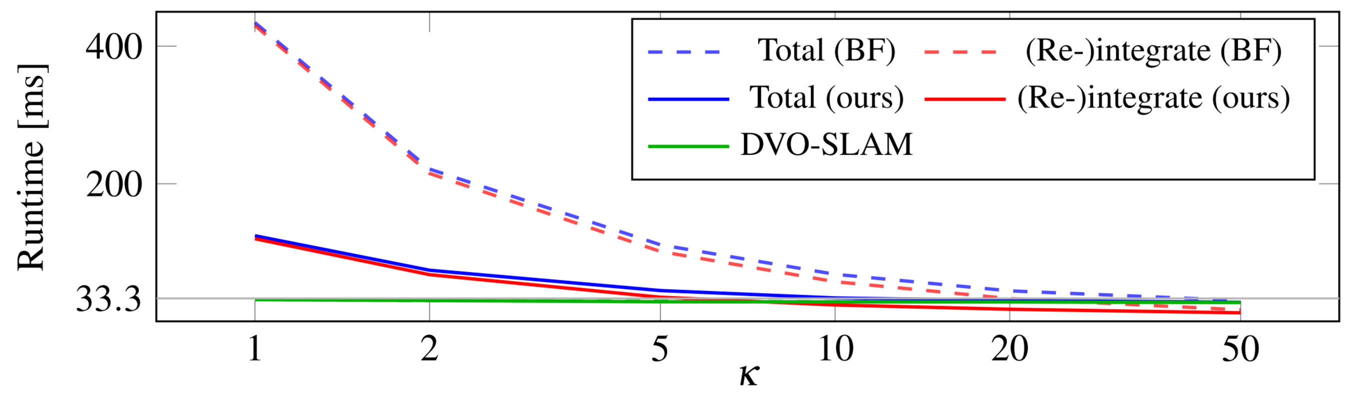


Correctness (k=5 vs. k=60) Completeness (k=5 vs. k=60)



Correctness Completeness

Runtime Evaluation



93% more efficient compared to BundleFusion [2] (k=20, m=5)

Qualitative Results: On-the-fly Surface Re-integration

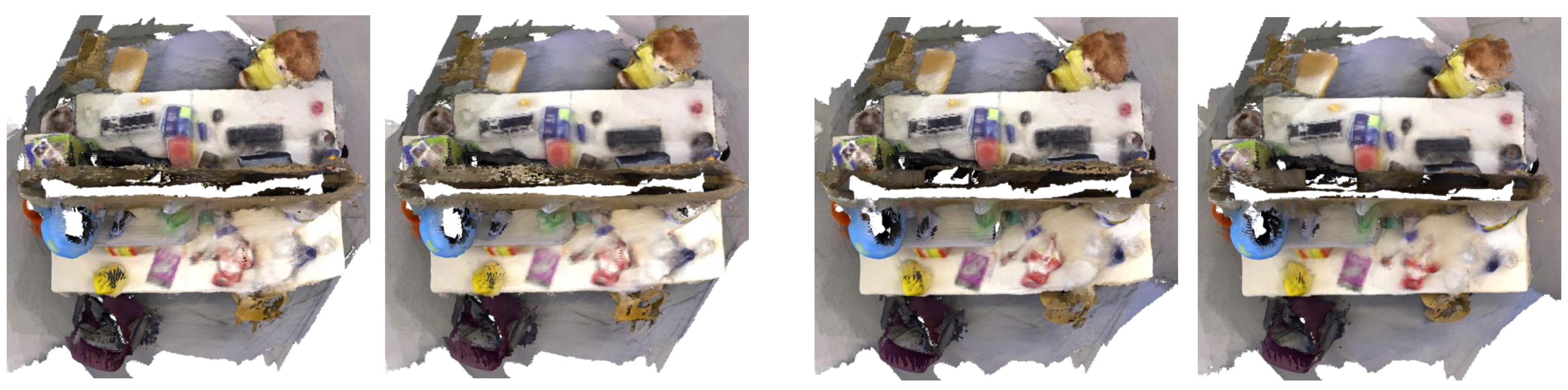
Without (left) and **with** (right) online surface correction

BundleFusion/apt0

i = 6000 i = 8000

Final

TUM/long_office_household



i = 1750 Final

Qualitative Comparison: ElasticFusion [3]

ElasticFusion [3] Ours

ElasticFusion [3] Ours

References

[1] C. Kerl, J. Sturm, D. Cremers. *Dense Visual SLAM for RGB-D Cameras.* IROS 2013.

[2] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, C. Theobalt. *BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration.* TOG 2017.

[3] T. Whelan, S. Leutenegger, R.F. Salas-Moreno, B. Glocker, A.J. Davison. *ElasticFusion: Dense SLAM Without A Pose Graph.* RSS 2015.