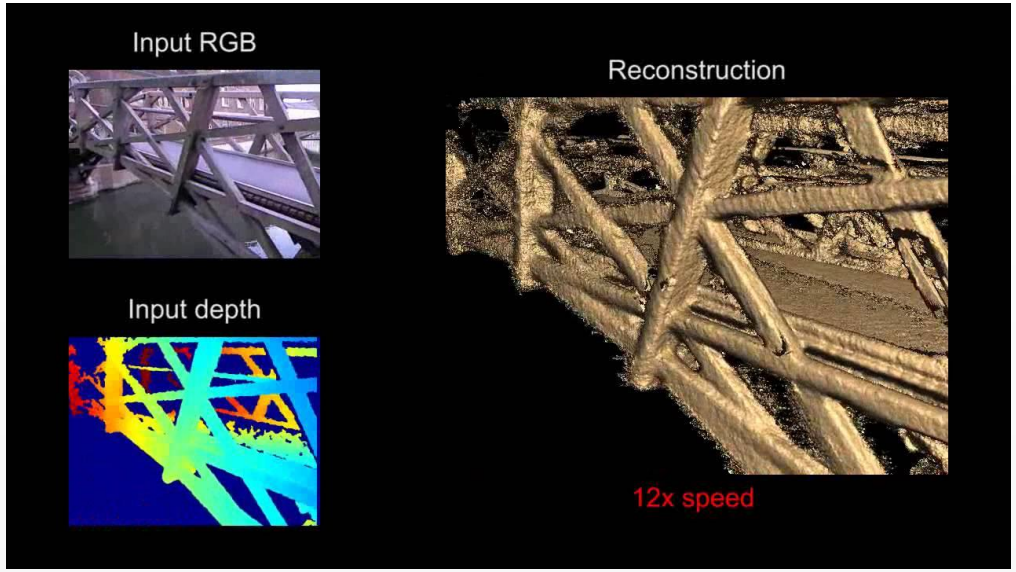
Feature-based RGB-D camera pose optimization for real-time 3D reconstruction

Chao Wang, Xiaohu Guo

University of Texas at Dallas

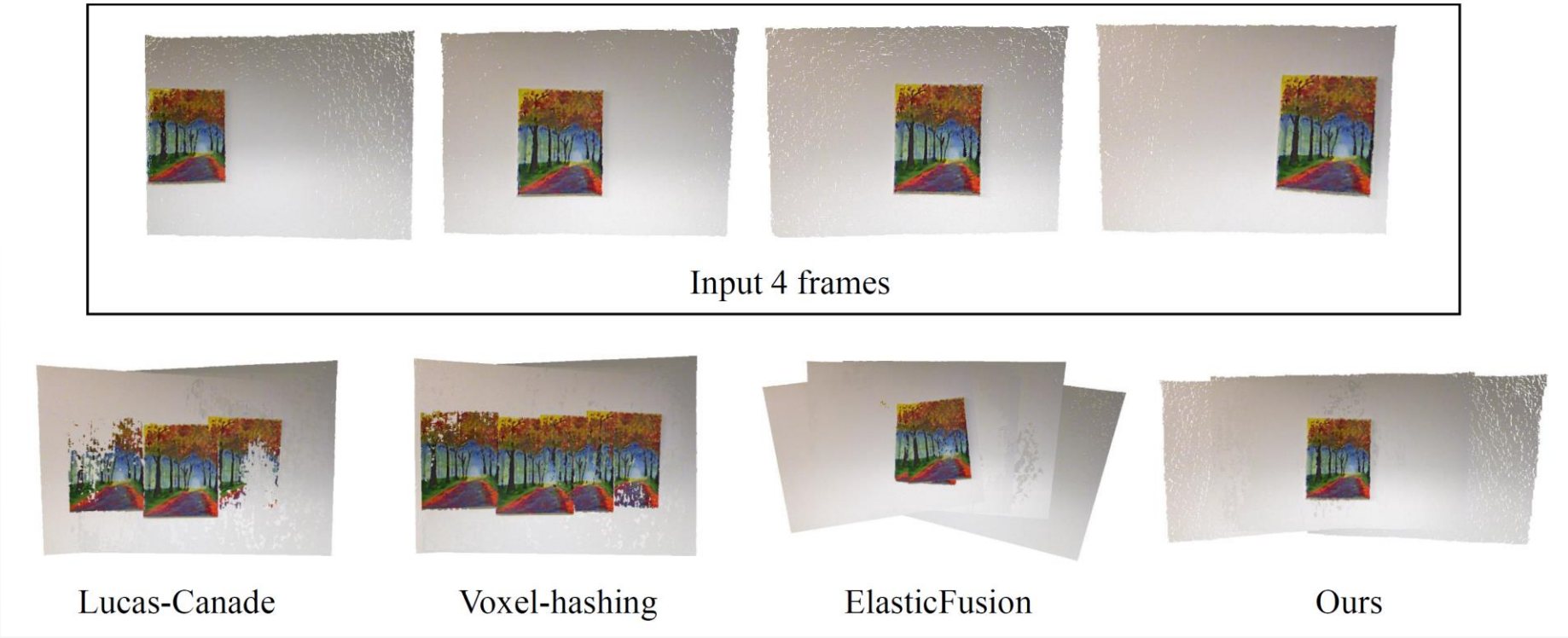
Camera pose estimation in online reconstruction

- ICP-based framework
  - Efficient and reliable with a good initial guess (small adjacent shifts)
  - Unreliable on data with large shifts, or large planar regions
  - KinectFusion [ISMAR11], Voxel-hashing [TOG13], ElasticFusion [RSS15], etc
- Feature-based estimation
  - Robust
  - Low accuracy: unreliable feature matching
  - Inefficient: most offline

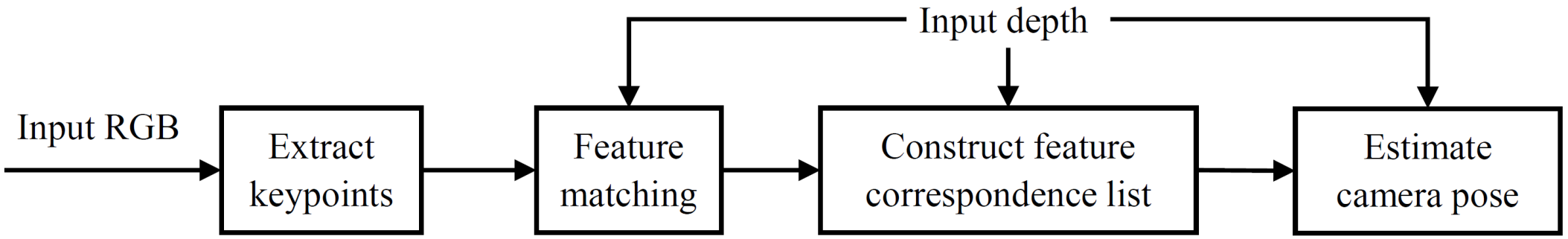


Our method

- Robust: improved feature-matching; feature correspondence list
- Efficient in real-time: low time complexity; GPU-acceleration

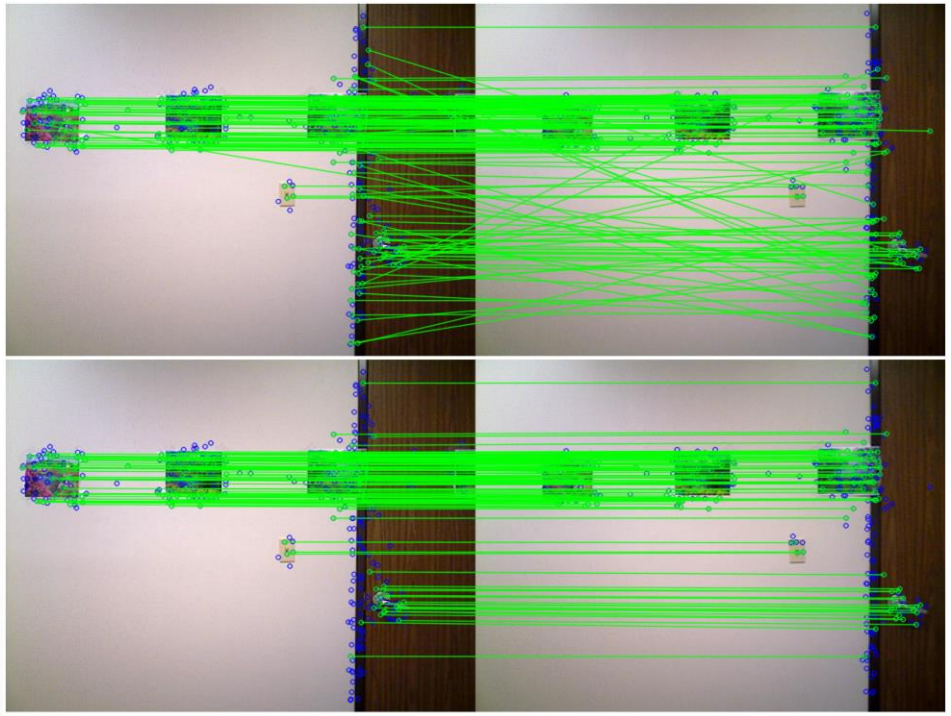


Outline

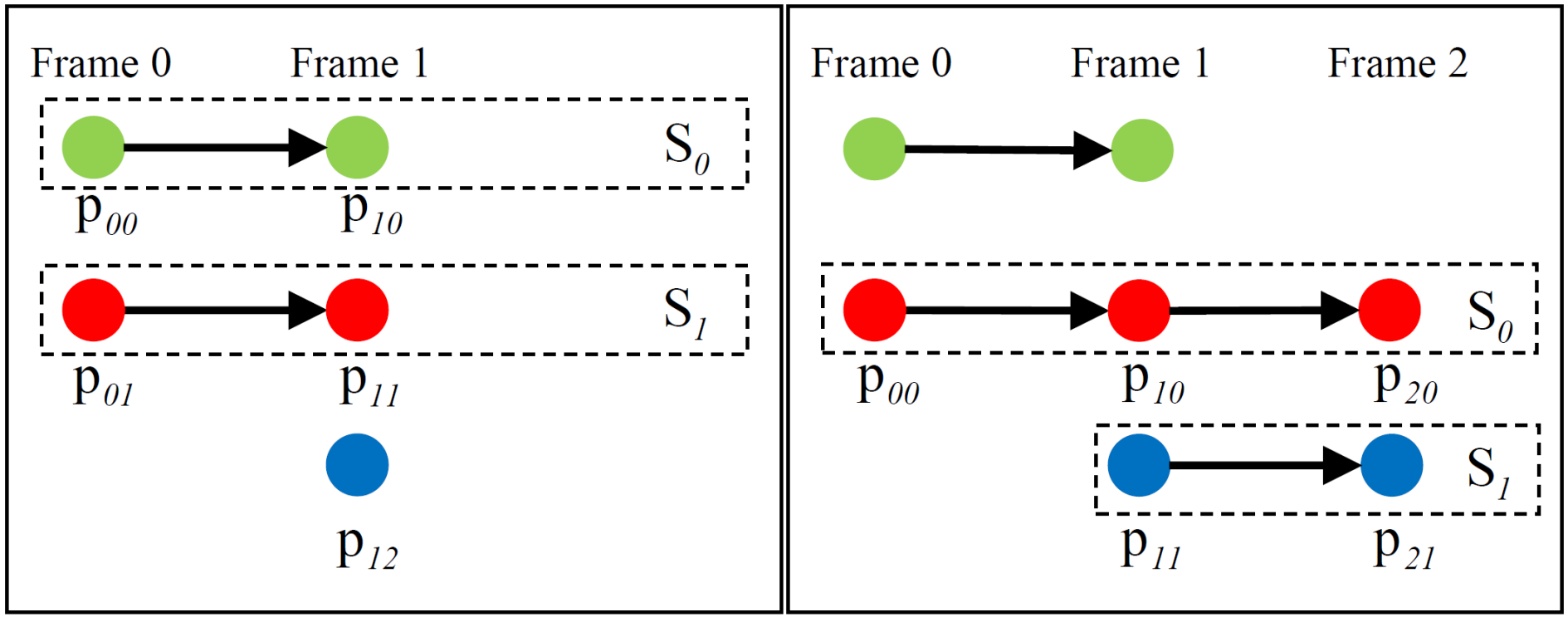

Input depth
Input RGB
Extract keypoints
Feature matching
Construct feature correspondence list
Estimate camera pose

Feature extraction and matching

- SURF: robust and efficient
- RANSAC-based correspondence check
  - Abandon unreliable depth
  - 2D homography
  - 3D relative transformation



Feature correspondence list (FCL) construction

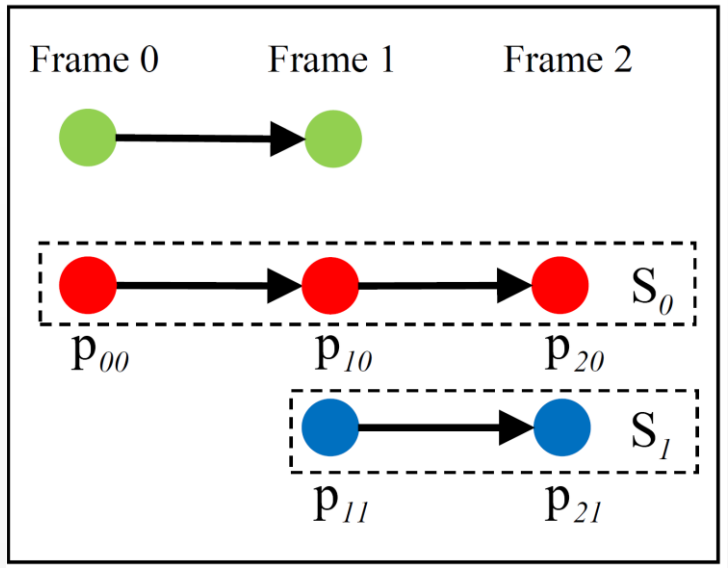


Camera pose optimization

$$E_i(\mathbf{R}_i, \mathbf{t}_i) = \sum_{j=0}^{m_i - 1} w_j \left\| \mathbf{R}_i \mathbf{p}_{ij} + \mathbf{t}_i - \mathbf{q}_j \right\|^2 , \qquad (1)$$

$$\mathbf{q}_j = \frac{1}{|\mathbf{S}_j| - 1} \sum_{k=n_j}^{i-1} \left(\mathbf{R}_k \mathbf{p}_{kj} + \mathbf{t}_k \right) , \qquad (2)$$

- i, j: frame index, and point index
- m_i: number of features in frame i
- p_{ij}: features' 3D points in camera space
- w_j: weight for point p_{ij}
- $\mathbf{R}_i, \mathbf{t}_i$: camera poses
- $\mathbf{q}_j$: mass center of jth point in global space (in previous frames)

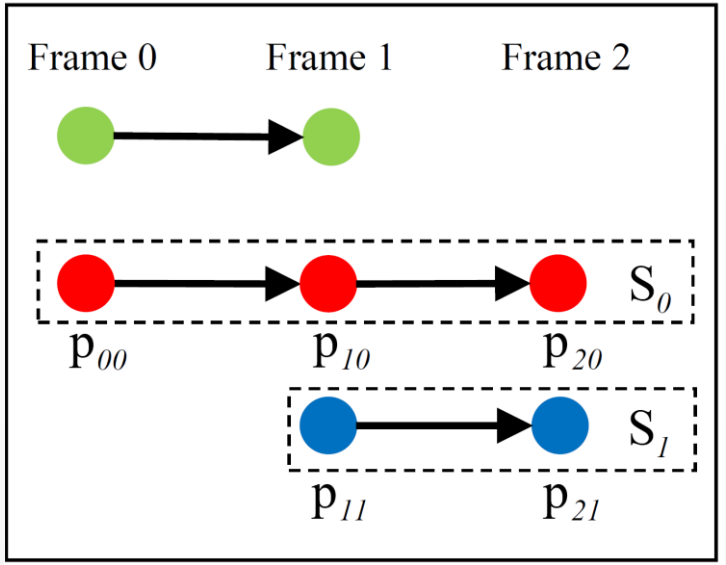


Camera pose optimization (cont.)

$$E(E_r, \ldots, E_i) = \sum_{k=r}^{i} E_i = \sum_{k=r}^{i} \sum_{j=0}^{m_k - 1} w_j \left\| \mathbf{R}_k \mathbf{p}_{kj} + \mathbf{t}_k - \mathbf{q}_j \right\|^2$$

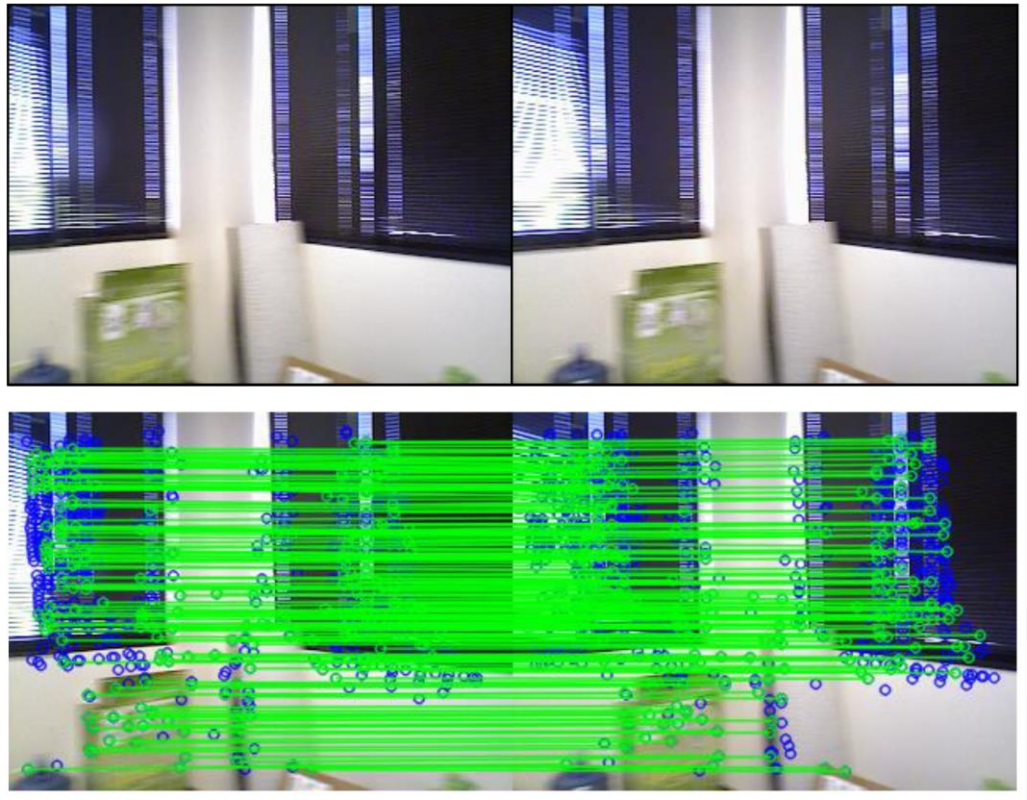
$$\mathbf{q}'_j = \frac{1}{|\mathbf{S}_j|} \sum_{k=n_j}^{i} (\mathbf{R}_k \mathbf{p}_{kj} + \mathbf{t}_k)$$

- i, j: frame index, and point index
- m_i: number of features in frame i
- p_{ij}: features' 3D points in camera space
- w_j: weight for point p_{ij}
- $\mathbf{R}_i, \mathbf{t}_i$: camera poses
- $\mathbf{q}_j$: mass center of jth point in global space (in previous frames)



Experimental Results

- Compared with Voxel-hashing [TOG13] and ElasticFusion [RSS15]
- Efficient: only 20ms per frame on average (5 ms in estimating pose)
- Robust on blurred images



Trajectory and pose estimation

Tab. 1 Trajectory estimation comparison using ATE metric between methods.

System	fr1/desk		fr1/floor		fr1/room		fr3/ntf	
	dif1	dif5	dif1	dif5	dif1	dif5	dif1	dif5
Voxel-hashing	1.10	0.74	1.01	0.70	0.61	1.08	1.32	1.30
Ours	0.32	0.32	**0.16**	**0.19**	0.34	0.61	0.18	0.082
ElasticFusion	**0.027**	0.30	0.41	0.54	0.38	0.48	**0.080**	0.15
Ours	0.035	**0.21**	0.17	0.22	**0.32**	**0.35**	**0.080**	**0.080**

Tab. 2 Pose estimation comparison using RPE metric between methods.

System	fr1/desk		fr1/floor		fr1/room		fr3/ntf	
	dif1	dif5	dif1	dif5	dif1	dif5	dif1	dif5
Voxel-hashing	1.57	1.16	1.25	0.98	1.15	1.49	1.60	1.62
Ours	0.91	0.94	0.80	0.80	0.84	1.14	1.36	1.42
ElasticFusion	**0.039**	0.41	**0.42**	0.58	**0.51**	0.63	**0.11**	**0.11**
Ours	0.045	**0.29**	0.43	**0.48**	0.54	**0.61**	0.12	**0.11**

Surface Reconstruction Comparison

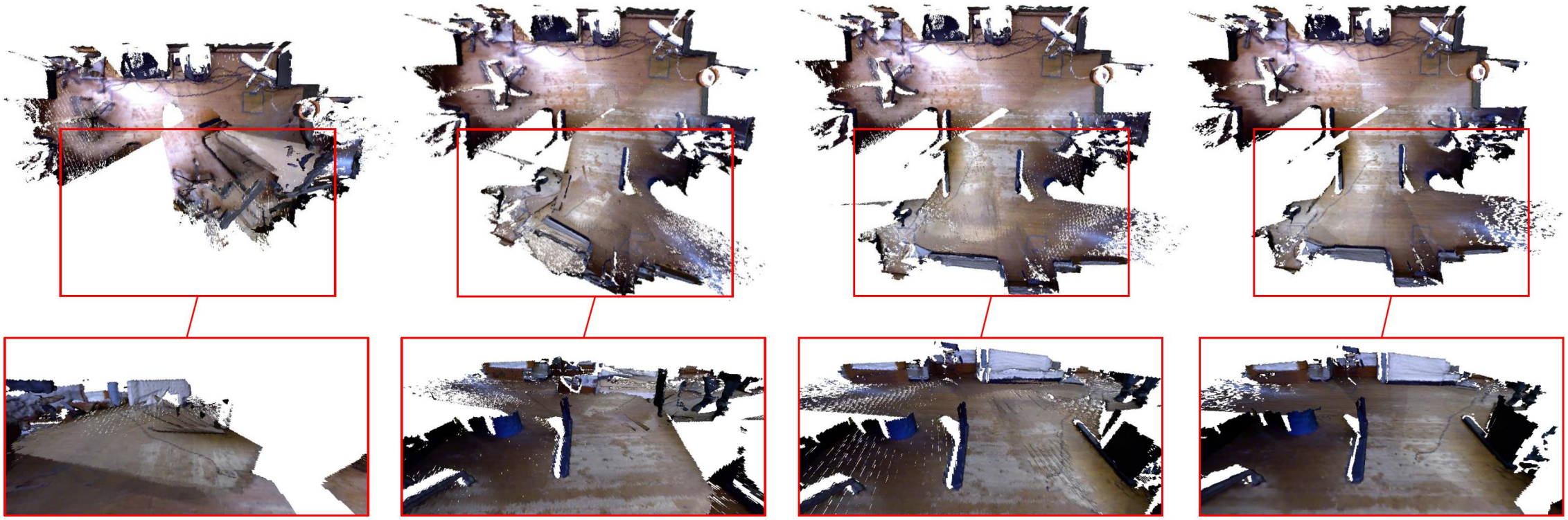
Data: fr1/floor

Frame difference: 5 (use 1 out of every 5 frames)

Note: here we show ElasticFusion's reconstruction result on Voxel-hashing platform by using its pre-computed camera poses on its original platform.

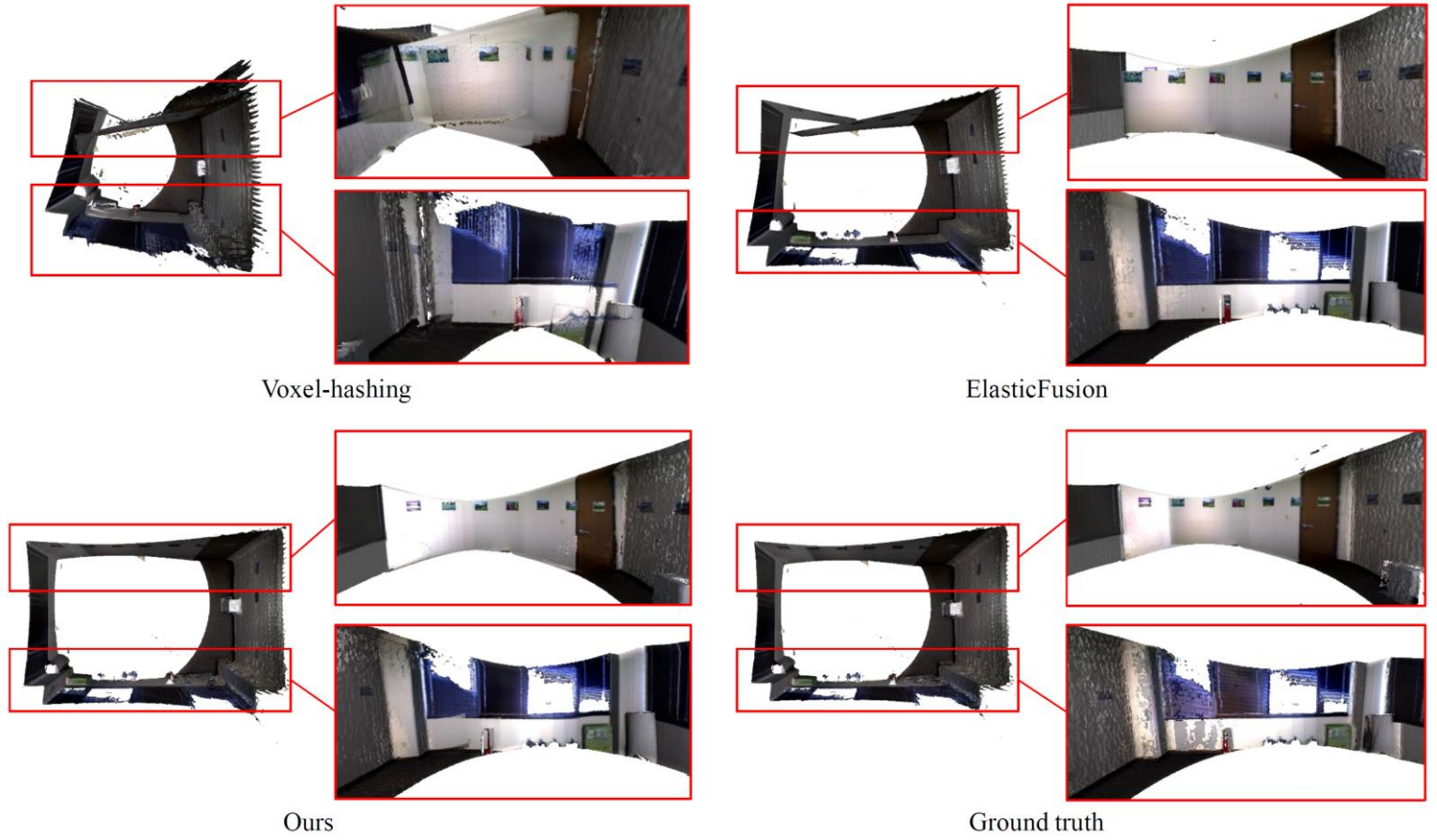
Reconstruction results on RGB-D benchmark

- fr1/floor [IROS12]



Reconstruction results on real scene

- Fast-scanned room: 235 frames



Conclusion

- Combination of two common strategies
- Robust on data with large adjacent shifts
- Efficient in real-time

Q & A

All questions are very welcome!