DeepTAM: Deep Tracking and Mapping

Huizhong Zhou* **Benjamin Ummenhofer*** Thomas Brox

{zhouh, ummenhof, brox}@cs.uni-freiburg.de

*equal contribution

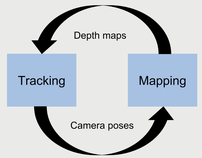
UNI FREIBURG — [o] Vision — COMPUTER VISION University of Freiburg

ECCV 2018 European Conference on Computer Vision

8 – 14 September 2018 | Munich, Germany

Contribution

Keyframe-based dense tracking and mapping with ConvNets



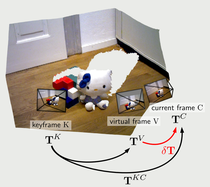
Contributions

- A tracking network architecture for incremental frame to keyframe tracking designed to reduce the dataset bias problem
- A multiple hypothesis approach for camera poses which leads to more accurate pose estimation.
- A mapping network architecture combining depth measurements with image-based priors, which is highly robust and yields accurate depth maps
- An efficient depth refinement strategy combining a network with a narrow band

Tracking Parameterization

Incremental parameterization

Instead of directly estimating the relative pose to the keyframe we estimate an increment to the previously tracked frame.



- Relative pose from keyframe to current frame $\mathbf{T}^{KC}$
- Pose increment from virtual to current frame $\delta\mathbf{T}$

Small pose changes are easier to learn than large pose changes!

Virtual keyframe

We render a virtual keyframe with the last pose estimate using the depth map and color image of the original keyframe. The rendered image relates the pose increment computation to the original keyframe.
During training we can place freely the virtual keyframe to simulate all motions.

Acknowledgements

This project was in large parts funded by the EU Horizon 2020 project Trimbot2020. We also thank the bwHPC initiative for computing resources, Facebook for their P100 server donation and gift funding.

Horizon 2020 European Union Funding for Research & Innovation

bw|HPC

facebook

Project Page

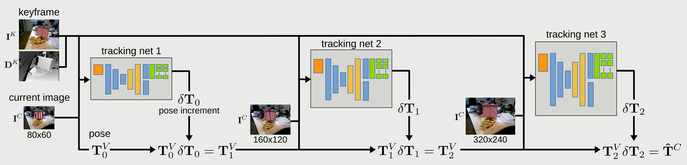
- **Paper**
- **Videos**
- **Code (coming soon)**

https://t1p.de/deeptam

Tracking Architecture

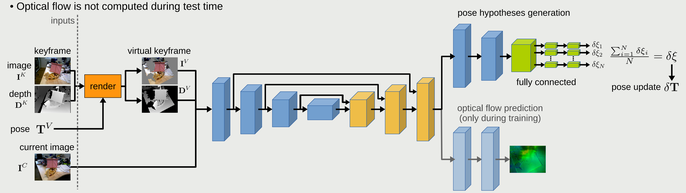
Coarse-to-fine scheme

We deal with large camera motions at smaller resolutions and then incrementally refine the pose at larger resolutions.
To this end we train 3 tracking networks with distinct parameters but similar architecture.



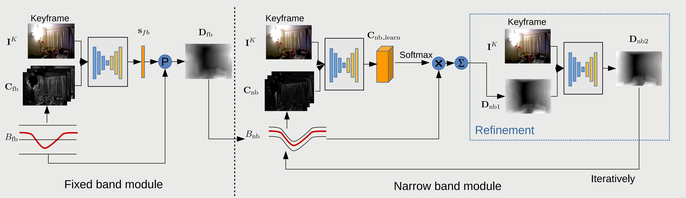
Tracking network design

- **Incremental pose**
  - Pose between keyframe and the rendered virtual frame
- **Multiple pose hypotheses**
  - Improves accuracy
  - Pose update as average of all hypotheses
- **Optical flow as auxiliary task**
  - Helps to stimulate the training of motion features and improves the performance
  - Optical flow is not computed during test time



Mapping Architecture

The mapping module estimates the keyframe depth from the keyframe image and the cost volume computed from a set of images and camera poses. It is divided into a fixed band module and a narrow band module.



Fixed band module

- Takes the keyframe image and the cost volume generated with 32 depth labels equally spaced in the depth range as inputs
- Extracts the depth estimate as an interpolation factor between the minimum and maximum depth label:

$$\mathbf{D}_{fb} = (1 - \mathbf{s}_{fb}) \cdot d_{min} + \mathbf{s}_{fb} \cdot d_{max}$$

Narrow band module

- Builds a cost volume around the current depth estimate with a certain band width
- A refinement network regularizes the depth map each iteration

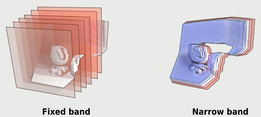
- Extracts the depth estimate using a differentiable soft argmin operation[7]:

$$\mathbf{D}_{nb1} = \sum_{d \in \mathbf{B}_{nb}} \mathbf{B}_{nb} \times \mathrm{softmax}(-\mathbf{C}_{nb,learn})$$

- Runs iteratively and gains in performance with more iterations

Cost Volume

Given a keyframe image and a sequence of frames with the corresponding relative camera poses, we can collect photometric information from multiple images in a cost volume. It stores the photoconsistency costs for each pixel at a set of depth labels.



Fixed band defines the depth labels equally spaced in a certain depth range:

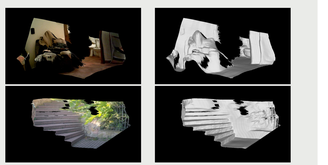
$$B_{fb} = \{b_i | b_i = d_{\min} + i \cdot \tfrac{d_{\max} - d_{\min}}{N-1}, i = 0, 1, ..., N-1\},$$

while narrow band centers around the previous depth estimate:

$$B_{nb} = \{b_i | b_i = d_{\text{prev}} + i \cdot \sigma_{nb} \cdot d_{\text{prev}}, i = -\tfrac{N}{2}, ..., \tfrac{N-2}{2}\}.$$

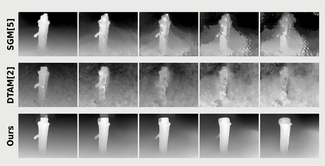
Meshes

Meshes generated from depth maps, which have been computed from sequences with 10 frames at a resolution of 320x240.



Mapping Robustness

Our mapping is more robust with respect to noisy camera poses than traditional methods. Even under large noise the depth map preserves important structures, which improves the robustness of the overall system.



Runtime

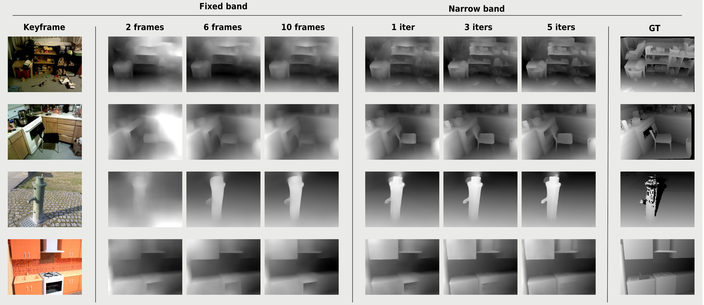
Runtime in seconds for the system components on a GTX 1070.
Tracking runs at ~44 Hz.
Mapping with 10 frames and 3 narrow band iterations runs at ~1.3Hz

	Tracking	Cost volume*	Fixed band	Narrow band**
Mean	0.0227	0.0164	0.0181	0.0359
Min	0.0203	0.0153	0.0171	0.0347
Max	0.0251	0.0168	0.0190	0.0393

* per frame
** per iteration

Mapping Results

The fixed band gains in performance with more frames but can also saturate. Based on the depth estimate of the fixed band, the narrow band network further increases the accuracy and captures more details.



Quantitative Comparison

Tracking evaluation on RGB-D benchmark[10]

We evaluate our tracking with the depth from the datasets and the depth estimated by our mapping.

	Tracking				Tracking and mapping	
Sequence	RGB-D SLAM Kerl et al.[8]	Ours (w/o flow)	Ours (w/o hypotheses)	Ours	CNN-SLAM* Tateno et al.[9]	Ours
fr1/360	0.125	0.069	0.065	**0.054**	0.500	**0.116**
fr1/desk	0.037	0.042	0.031	**0.027**	0.095	**0.078**
fr1/desk2	0.020	0.025	0.020	**0.017**	0.115	**0.055**
fr1/plant	0.062	0.063	0.060	**0.057**	**0.150**	0.165
fr1/room	0.042	0.051	0.041	**0.039**	0.445	**0.084**
fr1/rpy	0.082	0.070	**0.063**	0.065	0.261	**0.052**
fr1/xyz	0.051	0.030	0.021	**0.019**	0.206	**0.054**
average	0.060	0.050	0.043	**0.040**	0.253	**0.086**

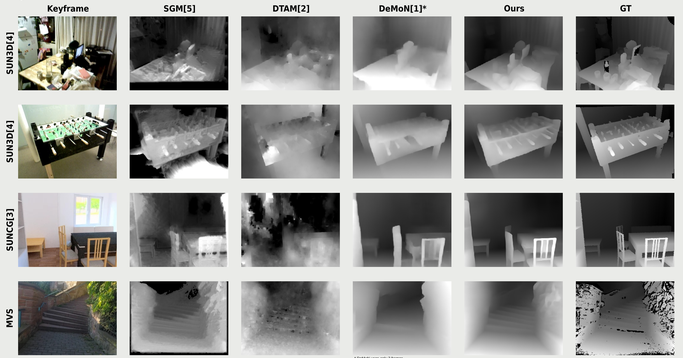
Mapping evaluation

We evaluate the influence of the number of frames and iterations for the fixed band and the narrow band, respectively. We compare favourably against other traditional and learning-based methods.

		Fixed band			Narrow band			Mapping comparison			
		2frames	6frames	10frames	1iter	3iters	5iters	SGM[5]	DTAM[2]	DeMoN[1]	Ours
SUN3D	L1-inv	0.117	0.085	**0.083**	0.076	0.065	**0.064**	0.210	0.197	-	**0.061**
	L1-rel	0.239	0.163	**0.159**	0.142	0.113	**0.111**	0.423	0.412	-	**0.111**
	sc-inv	0.193	0.169	**0.159**	0.156	0.132	**0.130**	0.374	0.340	0.146	**0.130**
MVS	L1-inv	0.075	**0.065**	0.067	0.049	0.039	**0.036**	0.086	0.059	-	**0.036**
	L1-rel	0.439	**0.418**	0.423	0.304	0.213	**0.171**	0.557	0.240	-	**0.171**
	sc-inv	0.213	**0.199**	0.200	0.174	0.152	**0.146**	0.305	0.246	0.251	**0.146**
SUNCG	L1-inv	0.097	0.067	**0.065**	0.050	**0.035**	0.036	0.142	0.169	-	**0.036**
	L1-rel	0.288	0.198	**0.193**	0.141	**0.082**	0.083	0.380	0.533	-	**0.083**
	sc-inv	0.206	0.174	**0.172**	0.155	**0.125**	0.128	0.343	0.383	0.248	**0.128**

Mapping Comparison

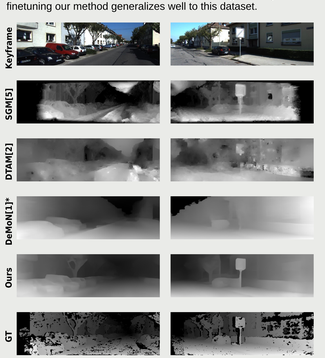
Qualitative depth prediction comparison for sequences with 10 frames. The classic methods have problems with short sequences and textureless scenes. DeMoN works well even in homogeneous image regions but misses many details. Our method can produce high quality depth maps using a small number of frames and captures more details.



* DeMoN uses only 2 frames

Generalization experiment on KITTI [6] w/o finetuning

KITTI is an urban scene dataset captured with a wide-angle camera, which differs from our training data significantly. Without finetuning our method generalizes well to this dataset.



Losses

Tracking

The training objective for the tracking network is

$$\mathcal{L}_{\text{tracking}} = \mathcal{L}_{\text{flow}}(\mathbf{w}) + \mathcal{L}_{\text{motion}}(\delta\xi) + \mathcal{L}_{\text{uncertainty}}(\delta\xi),$$

- Flow endpoint error: $\mathcal{L}_{\text{flow}} = \sum_{i,j} \|\mathbf{w}(i,j) - \mathbf{w}_{\text{gt}}(i,j)\|_2$
- L2 motion loss: $\mathcal{L}_{\text{motion}} = \alpha \|\mathbf{r} - \mathbf{r}_{\text{gt}}\|_2 + \|\mathbf{t} - \mathbf{t}_{\text{gt}}\|_2$
- Multivarariate laplace log likelihood:

$\mathcal{L}_{\text{uncertainty}} = \frac{1}{2}\log(|\mathbf{\Sigma}|) - 2\log\left(\frac{\mathbf{x}^\top\mathbf{\Sigma}^{-1}\mathbf{x}}{2}\right) - \log\left(K_\nu\left(\sqrt{2\mathbf{x}^\top\mathbf{\Sigma}^{-1}\mathbf{x}}\right)\right)$
with $\mathbf{x} = \delta\xi - \delta\xi_{\text{gt}}$ and $\mathbf{\Sigma} = \frac{1}{N}\sum_i^N (\delta\xi_i - \delta\xi)(\delta\xi_i - \delta\xi)^\top$ computed from the pose hypotheses.

Mapping

We use two losses on the depth maps.

- L1 depth loss: $\mathcal{L}_{\text{depth}} = |\mathbf{D} - \mathbf{D}_{\text{gt}}|$
- Scale invariant gradient loss from [1]:

$$\mathcal{L}_{\text{sc-inv-grad}} = \sum_{h \in \{1,2,4\}} \sum_{i,j} \|\mathbf{g}_h[\mathbf{D}](i,j) - \mathbf{g}_h[\mathbf{D}_{\text{gt}}](i,j)\|_2$$

with $\mathbf{g}_h[\mathbf{D}](i,j) = \left(\frac{\mathbf{D}(i+h,j)-\mathbf{D}(i,j)}{|\mathbf{D}(i+h,j)|+|\mathbf{D}(i,j)|}, \frac{\mathbf{D}(i,j+h)-\mathbf{D}(i,j)}{|\mathbf{D}(i,j+h)|+|\mathbf{D}(i,j)|}\right)^\top$

Example from [1]:

w/o sc-inv-grad — w/ sc-inv-grad

References

[1] B. Ummenhofer et al., "DeMoN: Depth and Motion Network for Learning Monocular Stereo," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[2] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in 2011 IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2320–2327.

[3] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic Scene Completion from a Single Depth Image," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 190–198.

[4] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels," in 2013 IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1625–1632.

[5] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, vol. 2, pp. 807–814 vol. 2.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 3354–3361.

[7] A. Kendall, H. Martirosyan, S. Dasgupta, and P. Henry, "End-to-End Learning of Geometry and Context for Deep Stereo Regression," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 66–75.

[8] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2100–2106.

[9] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[10] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 573–580.