UNI FREIBURG

[o] Vision COMPUTER VISION University of Freiburg

DAIMLER

DeMoN: Depth and Motion Network for Learning Monocular Stereo

Benjamin Ummenhofer, **Huizhong Zhou**, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, Thomas Brox

{ummenhof, zhouh, uhrigj, mayern, ilge, dosovits, brox}@cs.uni-freiburg.de

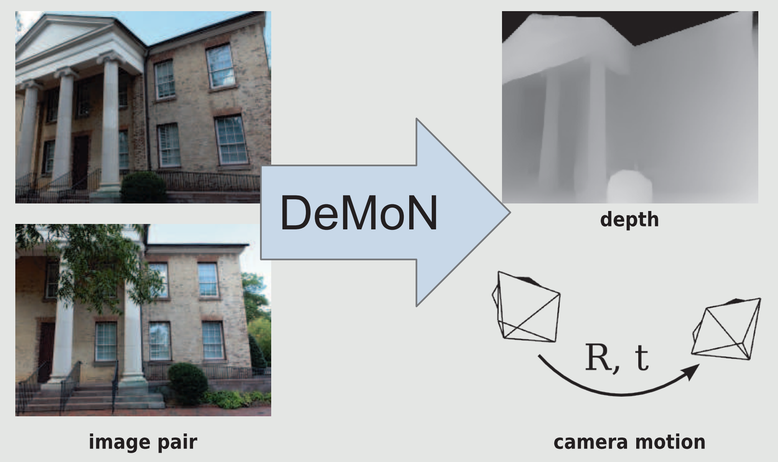
IEEE 2017 Conference on Computer Vision and Pattern Recognition

CVPR July 21-26 Honolulu 2017

Contribution

Two view geometry problem
Retrieving the camera motion and scene structure from two images is a fundamental problem in Structure from Motion (SfM).

DeMoN is a ConvNet architecture solving this problem.



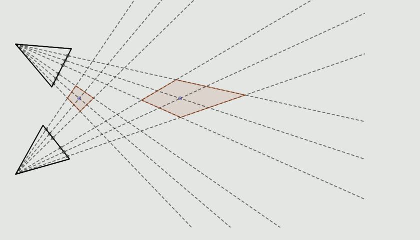
Contributions

- **A computer algorithm for reconstructing a scene from two projections**
- A network architecture exploiting motion parallax for depth prediction
- An iterative network part for refinement
- A scale invariant gradient loss for improved depth predictions
- Artificial datasets complementing shortcomings of real data

Depth & Motion Parameterization

Inverse depth
Depth uncertainty grows with increasing distance. Thus, we directly estimate the inverse depth (reciprocal of the depth values) to account for this.

- inverse depth $\xi = \frac{1}{z}$
- can represent points at infinity
- close objects are more important



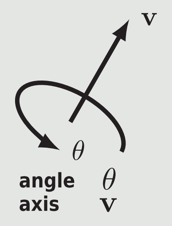
Motion
We present the camera motion from the first to the second frame as:

- 3D translation vector $\mathbf{t}$
- 3D angle axis vector $\mathbf{r} = \theta\mathbf{v}$

Angle axis representation

- Minimum parameterization

→ Network cannot generate invalid values



Scale ambiguity
Scene scale cannot be obtained from images in the **general case**. We resolve the ambiguity by normalizing translations such that

$$\|\mathbf{t}\| = 1$$

Estimated depth values need to correspond to the normalized translation. To facilitate adjusting the depth values we predict a scale factor along with the motion estimate and obtain $s\xi$.

Project Page

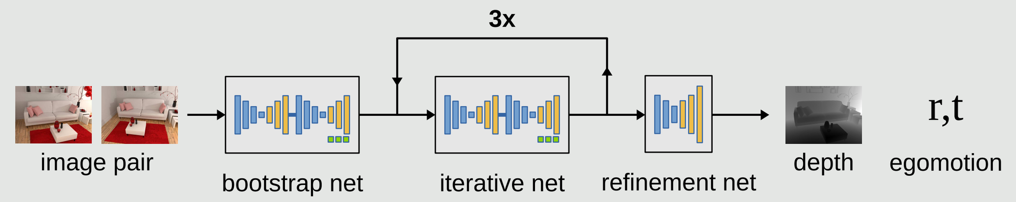
- **Paper**
- **Videos**
- **Code (Tensorflow)**

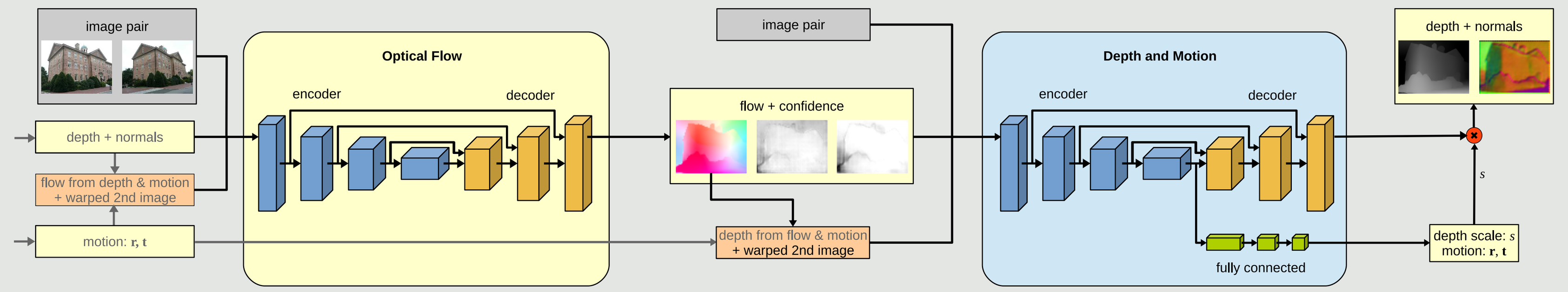
https://goo.gl/cXf4ct

Network Architecture

DeMoN consists of three subnets:

- **bootstrap net:** computes the initial depth and motion estimates
- **iterative net:** successively refines the previous estimates
- **refinement net:** increases the resolution of the final depth map





The bootstrap and iterative net use an encoder-decoder pair:

- **1st encoder-decoder:** estimates optical flow and its confidence
- **2nd encoder-decoder:** predicts depth and surface normals
- **a fully connected network appended to the 2nd encoder:** computes camera motion and a depth scale factor, which relates the scale of the depth values to the camera motion

Two images are better than one?!
A single encoder-decoder network does not make use of the second image and prefers to directly infer depth from a single image.

DeMoN explicitly solves the more difficult correspondence problem by computing optical flow in the first encoder-decoder.

Method	L1-inv	sc-inv	L1-rel
Single image	0.080	0.159	0.696
Naïve image pair	0.079	0.165	0.722
DeMoN	**0.012**	**0.131**	**0.097**

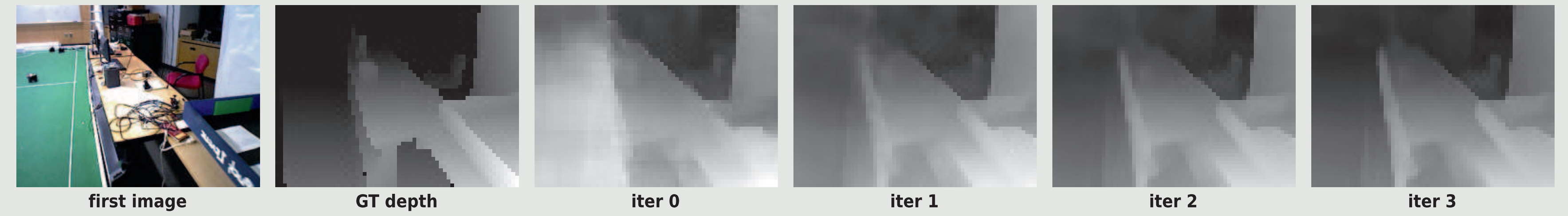
A naive architecture does not use the 2nd image

Iterative Refinement

The *iterative net* can improve and correct estimates from the *bootstrap net* or from previous iterations.

Wrong scale

Wrong depth



Iterative refinement on SUN3D
While we use 4 iterations during training, we find that 3 iterations on average gives the best results with respect to depth *and* motion.

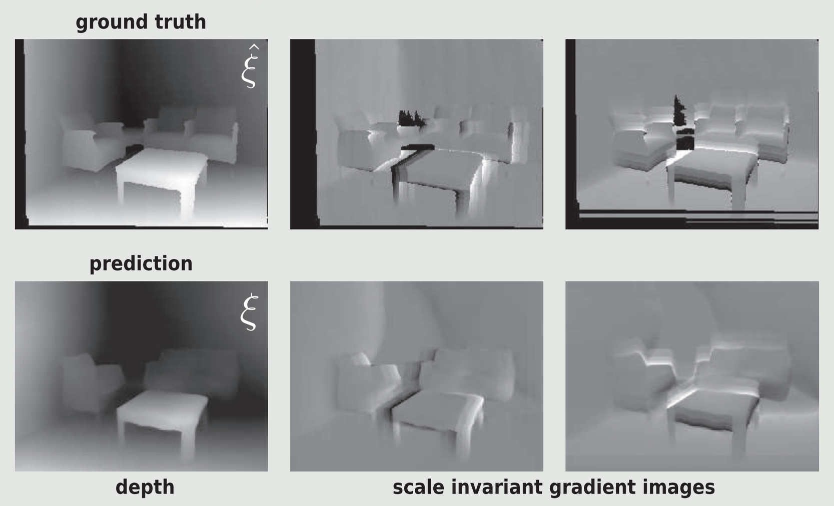
Performance slightly decays with many more iterations (>10) but remains stable.

Iteration	Depth L1-inv	Depth sc-inv	Depth L1-rel	Depth $\delta<1.25$	Depth $\delta<1.25^2$	Depth $\delta<1.25^3$	Motion rot	Motion tran	Flow EPE
0	0.029	0.145	0.244	0.587	0.844	0.940	2.18	20.27	0.030
1	0.024	0.130	0.207	0.679	0.891	0.961	1.94	17.25	0.020
2	0.022	0.131	0.187	0.688	0.900	0.982	1.87	18.31	0.019
3	0.021	0.132	0.179	0.698	0.912	0.981	1.80	18.81	0.019
4	0.021	0.133	0.184	0.690	0.908	0.975	1.79	18.94	0.019
5	0.021	0.133	0.185	0.692	0.910	0.970	1.79	19.65	0.019

Scale Invariant Gradient

Operator Definition
We define a finite differences operator invariant to scale changes:

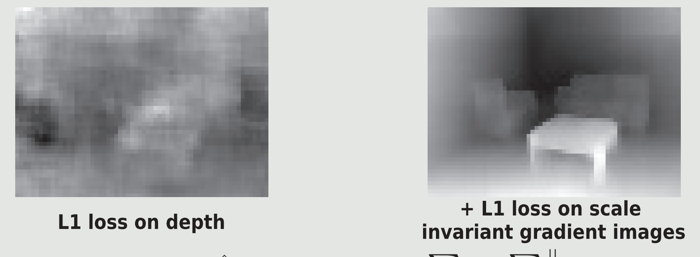
$$\mathbf{g}_h[f](i,j) = \left(\frac{f(i+h,j)-f(i,j)}{|f(i+h,j)|+|f(i,j)|}, \frac{f(i,j+h)-f(i,j)}{|f(i,j+h)|+|f(i,j)|} \right)^\top$$



Loss Design is Crucial!
In addition to L1 loss on the depth we compute a loss on the scale invariant gradient images (sig).

The loss on the sig images

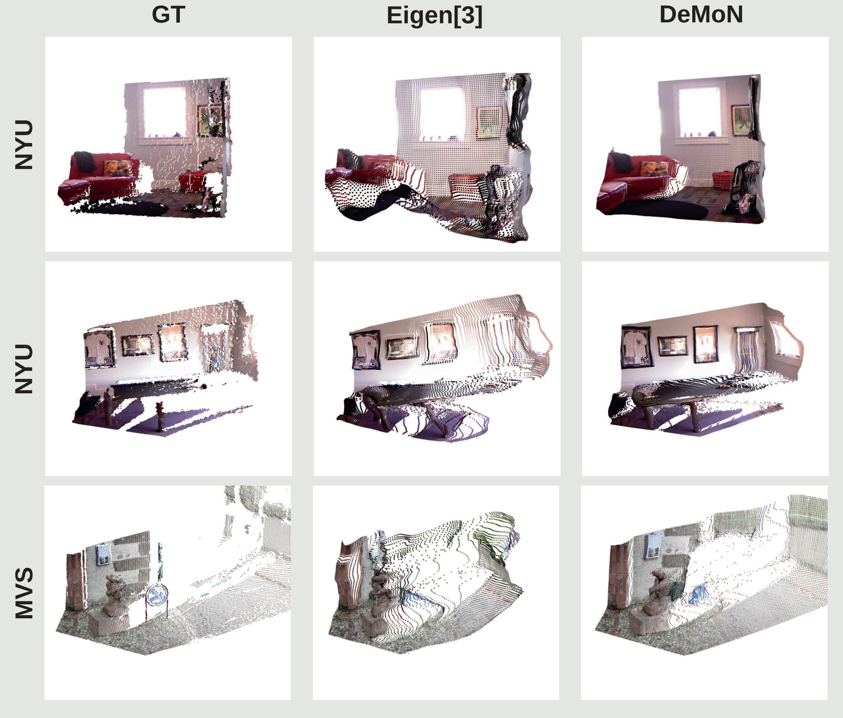
- emphasizes importance of depth discontinuities
- stimulates spatial comparisons



$$\mathcal{L}_{\text{depth}} = \sum_{i,j} |\xi(i,j) - \hat{\xi}(i,j)| \qquad \mathcal{L}_{\text{sig}\,\xi} = \sum_{h\in\{1,2,4,8,16\}} \sum_{i,j} \left\| \mathbf{g}_h[\xi](i,j) - \mathbf{g}_h[\hat{\xi}](i,j) \right\|_2$$

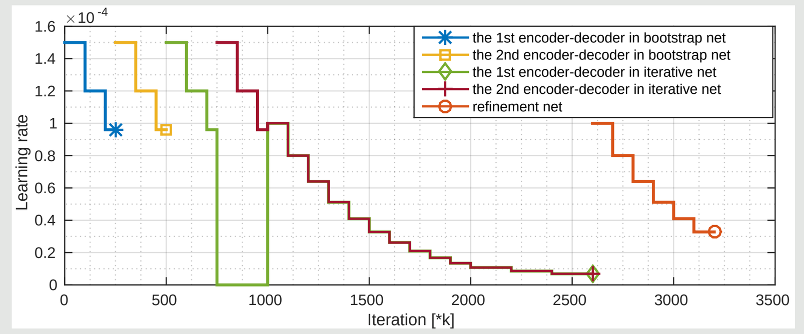
Point Cloud Comparison

Our method produces fewer depth artifacts, which can be seen if we visualize the depth as point clouds.



Training

The network training is based on the caffe[17] framework. It is trained from scratch on all the datasets jointly with Adam[18] using a momentum of 0.9 and a weight decay of 0.0004. We train sequentially the three subnets for 3200k iterations in total. A multistep learning rate policy is applied during training.



Flow Confidence

We train the flow confidence in a supervised manner. The ground truth confidence for the x component is given by the flow ground truth $\mathbf{w}_x(i,j)$ and the flow prediction $\hat{\mathbf{w}}_x(i,j)$:

$$\hat{c}_x(i,j) = e^{-|\mathbf{w}_x(i,j) - \hat{\mathbf{w}}_x(i,j)|}$$

Flow confidence helps the motion estimation since egomotion only requires sparse but high-quality correspondences.

Confidence	Motion rot	Motion tran	Flow EPE
no	2.830	25.262	0.027
yes	2.479	24.372	0.027

Datasets

Networks can easily overfit to training data, i.e. training on one dataset is not enough for a method trying to be as general as possible. We train on synthetic and real datasets with complementary properties to improve generalization.

Dataset	Perfect GT	Photorealistic	Outdoor scenes	Rot. avg	Rot. stddev	Tri. angle avg	Tri. angle stddev
SUN3D	no	yes	no	10.6	7.5	5.2	4.6
RGBD	no	yes	no	10.4	8.3	6.8	4.5
Scenes11	yes	no	(yes)	3.3	2.1	5.3	4.4
MVS	no	yes	yes	34.3	24.7	28.9	17.5
Blendswap	yes	(yes)	(yes)	23.1	17.1	20.1	13.6

SUN3D & RGBD

- Depth from structured light sensor
- Camera pose from SfM (SUN3D) or external tracking (RGBD)

Scenes11

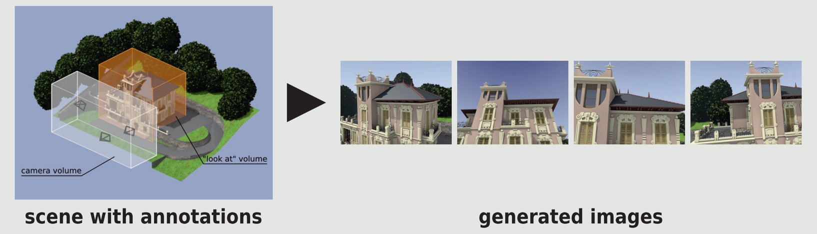
- Randomly generated scenes and objects from ShapeNet[2]

MVS

- Collection of Multi VIew Stereo datasets
- Depth and camera poses from SfM pipelines[4,10,11,16]

Blendswap

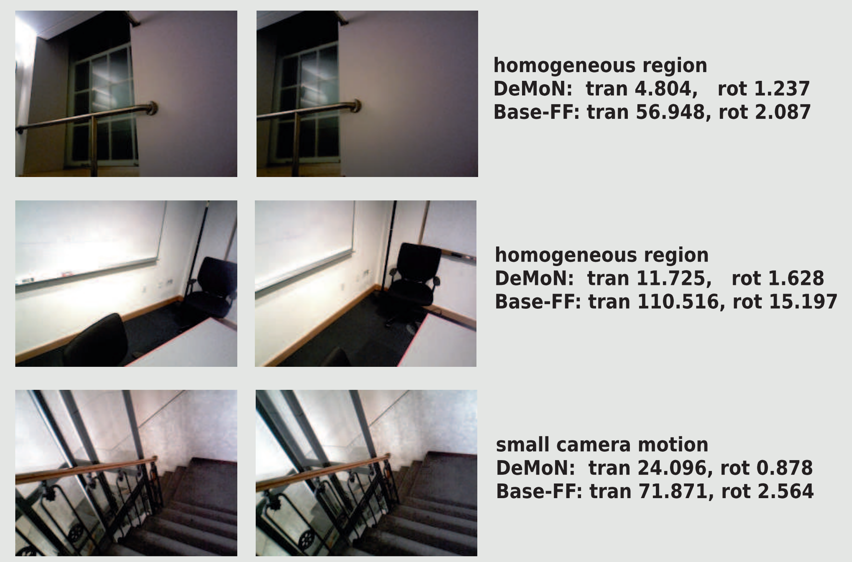
- About 150 distinct scenes from `blendswap.com`
- Annotated to enable automatic generation of image pairs



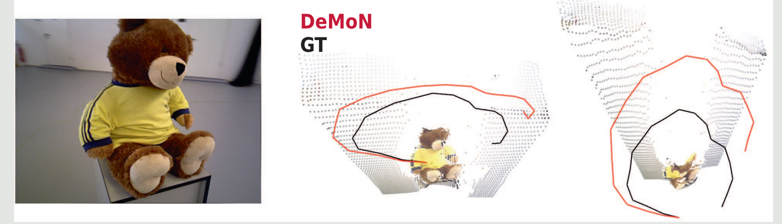
Motion

Our method can estimate the camera motion in scenarios difficult for traditional approaches like low texture or small motions.

Failure cases for Base-FF

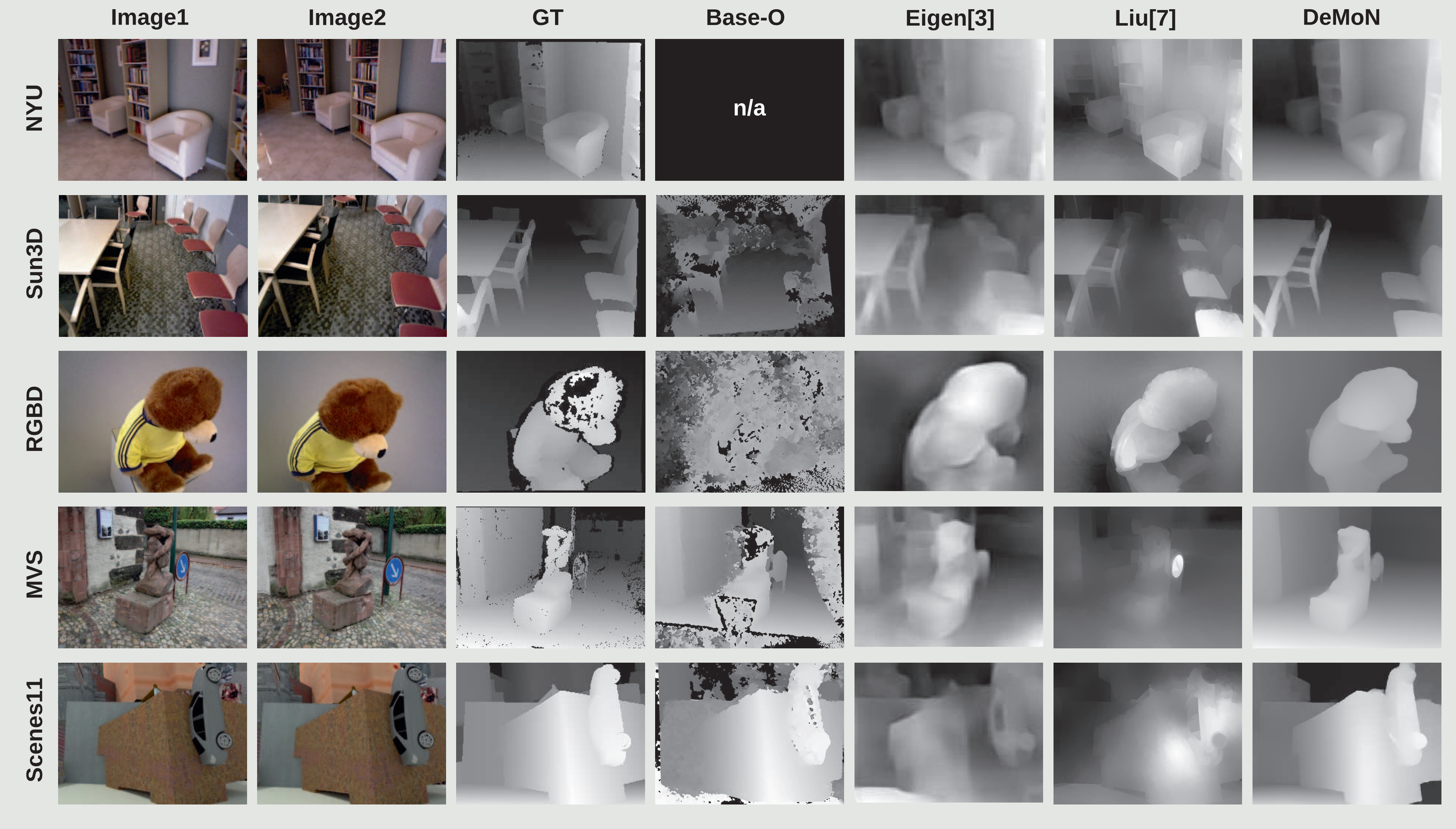


Concatenated pairwise motions
The local pairwise camera poses are consistent with the ground truth.

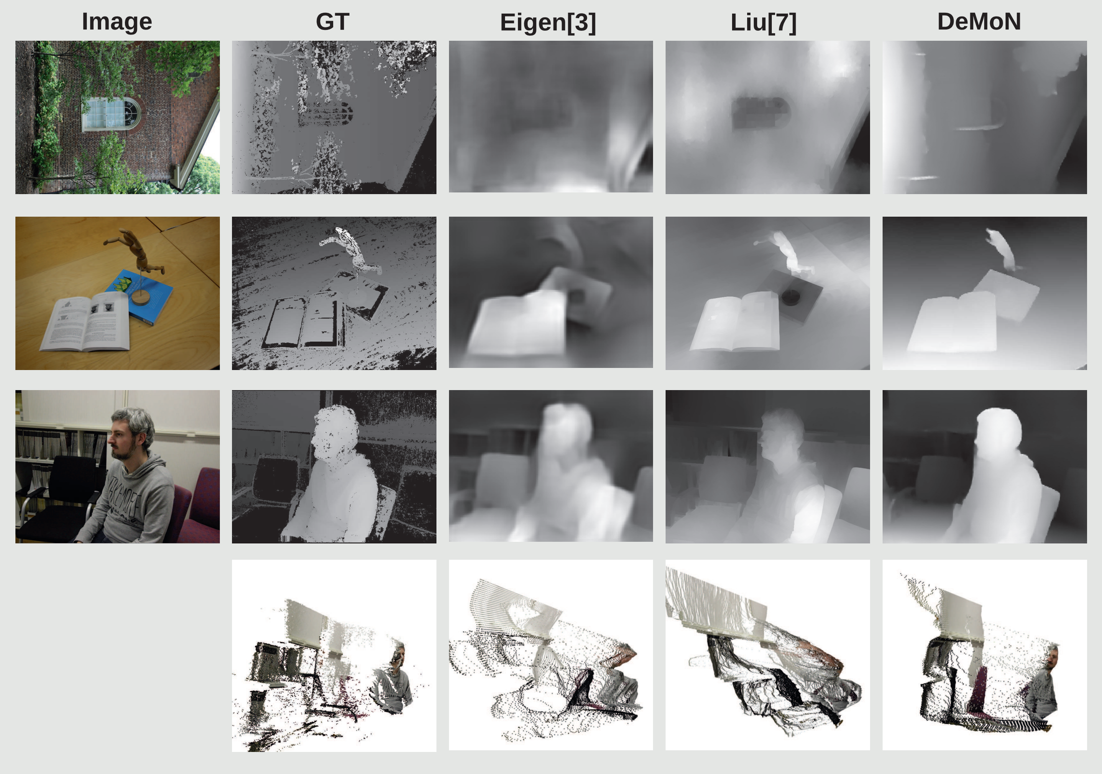


Depth Comparison

The depth maps produced by DeMoN are more detailed and more regular than the ones produced by other methods.



Generalization to new data
DeMoN exploits the geometric relations between a pair of images and therefore generalizes better to unknown scenes for example close-ups of people and objects, images rotated by 90 degrees .



Quantitative Comparison

We compare against several traditional methods as well as CNN single image methods.

- DeMoN outperforms all baseline methods on most datasets.
- Besides visual quality, we quantitatively perform as good or better than the single image methods.

	Correspondences	E-Matrix Method	Depth Method
Base-Oracle	-	-	SGM NCC[6]
Base-SIFT	SIFT[8]	8-point algorithm[5]	SGM NCC[6]
Base-FF	Flowfields[1]	8-point algorithm[5]	SGM NCC[6]
Base-Matlab	KLT[12,15]	5-point algorithm[9]	-
Base-Mat-F	DeMoN	5-point algorithm[9]	-

Classic methods:

Dataset	Method	Depth L1-inv	Depth sc-inv	Depth L1-rel	Motion rot	Motion trans
MVS	Base-Oracle	**0.019**	**0.197**	**0.105**	0	0
	Base-SIFT	0.056	0.309	0.361	21.180	60.516
	Base-FF	0.055	0.308	0.322	**4.834**	17.252
	Base-Matlab	-	-	-	10.843	32.736
	Base-Mat-F	-	-	-	5.442	18.549
	DeMoN	0.047	0.202	0.305	5.156	**14.447**
Scenes11	Base-Oracle	0.023	0.618	0.349	0	0
	Base-SIFT	0.051	0.900	1.027	6.179	56.650
	Base-FF	0.038	0.793	0.776	1.309	19.425
	Base-Matlab	-	-	-	0.917	14.639
	Base-Mat-F	-	-	-	2.324	39.055
	DeMoN	**0.019**	**0.315**	**0.248**	**0.809**	**8.918**
RGB-D	Base-Oracle	**0.026**	0.398	0.336	0	0
	Base-SIFT	0.050	0.577	0.703	12.010	56.021
	Base-FF	0.045	0.548	0.613	4.709	46.058
	Base-Matlab	-	-	-	12.831	49.612
	Base-Mat-F	-	-	-	2.917	22.523
	DeMoN	0.028	**0.130**	**0.212**	**2.641**	**20.585**
Sun3D	Base-oracle	0.020	0.241	0.220	0	0
	Base-SIFT	0.029	0.290	0.286	7.702	41.825
	Base-FF	0.029	0.284	0.297	3.681	33.301
	Base-Matlab	-	-	-	5.920	32.298
	Base-Mat-F	-	-	-	2.230	26.338
	DeMoN	**0.019**	**0.114**	**0.172**	**1.801**	**18.811**
NYUv2	Base-oracle	-	-	-	-	-
	Base-SIFT	-	-	-	-	-
	Base-FF	-	-	-	-	-
	Base-Matlab	-	-	-	-	-
	Base-Mat-F	-	-	-	-	-
	DeMoN	-	-	-	-	-

Single image methods:

Dataset	Method	Depth sc-inv
MVS	Liu indoor	0.260
	Liu outdoor	0.341
	Eigen VGG	0.225
	DeMoN	**0.203**
Scenes11	Liu indoor	0.816
	Liu outdoor	0.814
	Eigen VGG	0.763
	DeMoN	**0.303**
RGB-D	Liu indoor	0.338
	Liu outdoor	0.428
	Eigen VGG	0.272
	DeMoN	**0.134**
Sun3D	Liu indoor	0.214
	Liu outdoor	0.401
	Eigen VGG	0.175
	DeMoN	**0.126**
NYUv2	Liu indoor	0.210
	Liu outdoor	0.421
	Eigen VGG	**0.148**
	DeMoN	0.180

classic methods **single image methods**

References

[1] C. Bailer, B. Taetz, and D. Stricker, "Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation," in Computer Vision (ICCV). International Conference on Computer Vision (ICCV-15), December 13-16, Santiago, Chile, 2015.
[2] A. X. Chang et al., "ShapeNet: An Information-Rich 3D Model Repository," arXiv:1512.03012 [cs], Dec. 2015.
[3] D. Eigen and R. Fergus, "Predicting Depth, Surface Normals and Semantic Labels With a Common Multi-Scale Convolutional Architecture," in The IEEE International Conference on Computer Vision (ICCV), 2015.
[4] S. Fuhrmann, F. Langguth, and M. Goesele, "MVE: A Multi-view Reconstruction Environment," in Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage, Aire-la-Ville, Switzerland, Switzerland, 2014, pp. 11–18.
[5] R. I. Hartley, "In defense of the eight-point algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pp. 580–593, Jun. 1997.
[6] 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.
[7] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," arXiv:1502.07411 [cs], Feb. 2015.
[8] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[9] D. Nister, "An efficient solution to the five-point relative pose problem," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pp. 756–770, Jun. 2004.
[10] J. L. Schönberger and J. M. Frahm, "Structure-from-Motion Revisited," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4104–4113.
[11] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise View Selection for Unstructured Multi-View Stereo," in Computer Vision – ECCV 2016, 2016, pp. 501–518.
[12] J. Shi and C. Tomasi, "Good features to track," in 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.
[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in Computer Vision – ECCV 2012, 2012, pp. 746–760.
[14] 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.
[15] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," International Journal of Computer Vision, 1991.
[16] C. Wu, "Towards Linear-Time Incremental Structure from Motion," in 2013 International Conference on 3D Vision - 3DV 2013, 2013, pp. 127–134.
[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S.Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093, 2014.
[18] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs], Dec. 2014. arXiv: 1412.6980.
[19] J. Xiao, A. Owens, and A. Torralba. SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels. In IEEE International Conference on Computer Vision (ICCV), pages 1625–1632, Dec. 2013.

Acknowledgements

We acknowledge funding by the ERC Starting Grant VideoLearn, the DFG grant BR-3815/5-1 and the EU project Trimbot2020.

DFG erc SEVENTH FRAMEWORK PROGRAMME Horizon 2020 European Union funding for Research & Innovation