GLOBAL-LOCAL NETWORK FOR LEARNING DEPTH VIA INTERACTION

Antonio Loquercio[1], **Alexey Dosovitskiy**[2], **Davide Scaramuzza**[1],
Dept. Informatics, UZH and Dept. of Neuroinformatics, UZH and ETH Zurich[1],
Google Brain, Berlin[2]

ABSTRACT

How do babies learn to perceive the three-dimensional structure of the world? Motivated by the astonishing capabilities of natural intelligent agents and inspired by theories from psychology, this paper explores the idea that perception gets coupled to 3D properties of the world via interaction with the environment. Existing works for depth estimation require either massive amounts of annotated training data or some form of hard-coded geometrical constraint. This paper explores a new approach to learning depth perception requiring neither of those. Specifically, we make use of training data similar to what would be available to an agent interacting with the environment via haptic feedback: very sparse depth measurements, just a few pixels per image. To learn from such extremely sparse supervision, we design a specialized global-local network architecture that takes a pair of images and outputs a latent representation of the observer's motion between the images and a dense depth map.

1 INTRODUCTION

Understanding of the three-dimensional structure of the world is crucial for the functioning of intelligent agents: for instance, it supports path planning and navigation, as well as motion planning and object manipulation. Animals, including humans, obtain such three-dimensional understanding naturally, without any specialized training. By simultaneously observing the environment and interacting with it (1), they learn to estimate distances to objects using stereopsis and a variety of monocular cues (2; 3), including motion parallax, perspective, defocus, familiar object sizes. How could artificial systems acquire similar spatial awareness?

This question inspired a long line of work on algorithmically extracting three-dimensional structures from their two-dimensional projections (4). Classically, multi-view geometry is used to reconstruct the 3D coordinates of points given their corresponding projections in multiple images. One downside of this class of approaches is that they are using only some of the depth cues (mainly, stereo and motion parallax), but typically do not exploit more subtle monocular cues, such as perspective, defocus or familiar object size. Unsupervised learning approaches to depth estimation (5; 6) combine geometry with deep learning, with the hope that deep networks can learn to utilize the cues not used by the classic methods. Unsupervised learning approaches are remarkably successful in many cases, but they are fundamentally based on hard-coded geometry equations, which makes them biologically implausible and potentially sensitive to the precise specification of the camera model and parameters.

The present work is motivated by the following question: how can three-dimensional perception be learned by an embodied agent with a general learning algorithm without the explicit use of projective geometry? At the first glance this may seem like a virtually impossible task: how can two-dimensional images of the environment be connected to the metric properties of the scene without the use of 3D geometry? To make the problem tractable, we make two assumptions. First, motivated by the extensive evidence from psychology and neuroscience on the fundamental importance of motion perception (7; 8; 9; 10), we provide pre-computed optical flow as an input to the depth estimation system. Optical flow estimation can be learned either from synthetic data (11; 12; 13) or from real data in an unsupervised fashion (14; 15). Second, inspired by theories from psychology (1), we explore the idea that perception gets coupled to 3D properties of the world via interaction with the environment. In particular, we assume the availability of very sparse depth ground truth, just several

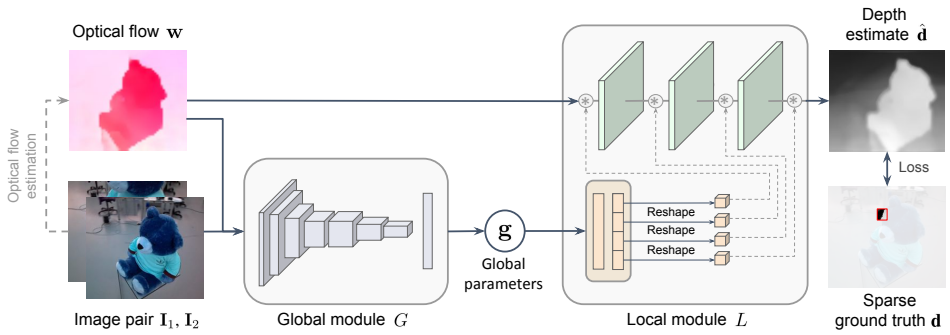

Figure 1: Global-local model architecture. An image pair and an estimated flow field are first fed through the global module that estimates the "global parameters" vector $\mathbf{g}$, representing the camera motion. From these global parameters, the local module generates three convolutional filter banks and applies them to the optical flow field. The output of the local module is then processed by a convolution to generate the final depth estimate.

pixels per image, similar to what an agent might collect by touching objects in the environment or bumping into obstacles.

In order to learn from such sparse annotations, we design a lightweight global-local network architecture (see Fig. 1) consisting of two modules – global and local – inspired by camera pose estimation and triangulation in standard geometric pipelines. We demonstrate that our network learns to estimate depth when provided, at training time, ground-truth for as little as a single pixel per image, i.e., 0.002% of the agent's field of view.

2 Related Work

The problem of recovering the three-dimensional structure of a scene from its two-dimensional projections has been long studied in computer vision (16; 17; 18; 19). Classic methods are based on multi-view projective geometry (4).

Supervised learning methods have demonstrated impressive results (20; 21; 22; 23) when trained on large amount of data. However, it is desirable to develop algorithms that function in the absence of such large annotated datasets. Unsupervised (or self-supervised) learning provides an attractive alternative to the label-hungry supervised learning. The dominant approach is inspired by classic 3D reconstruction techniques and makes use of projective geometry and photometric consistency across frames. Several works, similar to ours, aim to learn 3D representations without explicitly applying geometric equations (24; 25; 26). A scene, represented by one or several images, is encoded by a deep network into a latent vector, from which, given a target camera pose, a decoder network can generate new views of the scene. A downside of this technique is that the 3D representation is implicit and therefore cannot be directly used for downstream tasks such as navigation or motion planning. Moreover, at training time it requires knowing camera pose associated with each image. Our method, in contrast, does not require camera poses, and grounds its predictions in the physical world via very sparse depth supervision. This allows us to learn an explicit 3D representation in the form of depth maps.

3 Methodology

Given two monocular RGB images $\mathbf{I}_1$, $\mathbf{I}_2$, with unknown camera parameters and relative pose, as well as the optical flow $\mathbf{w}$ between them, we aim to estimate a dense depth map corresponding to the first image. We assume to have an artificial agent equipped with a range sensor, which navigates through an indoor environment. By doing so, it collects a training dataset of image pairs, with depth ground truth $\mathbf{d}$ available only for extremely few pixels. We now describe the network architecture.

3.1 MODEL ARCHITECTURE

An overview of the global-local network architecture is provided in Figure 1. The system operates on an image pair $\mathbf{I}_1, \mathbf{I}_2$ and the optical flow (dense point correspondences) $\mathbf{w}$ between them. In this work, we estimate the flow field with an off-the-shelf optical flow estimation algorithm, which is neither trained nor tuned on our data.

The rest of the model is composed of two modules: a global module G that processes the whole image and outputs a compact vector of "global parameters" and a local module L that applies a compact fully convolutional network, conditioned on the global parameters, to the optical flow field. This design is motivated both by classic 3D reconstruction methods and by machine learning considerations. Establishing an analogy with classic pipelines, the global module corresponds to the relative camera pose estimation, while the local module corresponds to triangulation – estimation of depth given the image correspondences and the camera motion. These connections are described in more detail in the supplement. From the learning point of view, we aim to train a generalizable network with few labels, and therefore need to avoid overfitting. The local module is very compact and operates on a transferable representation – optical flow. The global network is bigger and takes raw images as input, but it communicates with the rest of the model only via the low-dimensional bottleneck of global parameters, which prevents potential overfitting.

Similarly to previous work (22; 21), we define the loss on the inverse depth $\hat{\mathbf{z}} \doteq \hat{\mathbf{d}}^{-1}$. This is a common representation in computer vision and robotics (27; 28), which allows to naturally handle points and their uncertainty over a large range of depths.

4 EXPERIMENTS

We test the approach on three datasets collected in cluttered indoor environments, either real or simulated: Scenes11 (22), SUN3D (29), and RGB-D SLAM (30). For all datasets, we use the splits proposed by (22). In order to simulate range observations by the agent, we mask out all depth ground truth except for a single pixel (unless mentioned otherwise). As commonly done in two-view depth estimation methods (22) and in structure-from-motion methods (31), we resolve the inherent scale ambiguity by normalizing the depth values such that the norm of the translation vector between the two views is equal to 1. To quantitatively evaluate the generated depth maps, we adopt the three standard error metrics proposed by (22).

4.1 LEARNING FROM VERY SPARSE GROUND TRUTH

We compare the proposed global-local architecture to strong generic deep models – the encoder-decoder architecture of Eigen et al. (20), the popular fully convolutional architecture DispNet (32), and the multi-scale encoder-decoder of Laina et al. (FCRN) (33). Note that for a fair comparison with our method, we provide all the baselines with both the image pair and the optical flow field. We additionally tune the models to reach best performance on our task. The details of the tuning process are reported in Section 6.2.1 in the Appendix. We also compare to a reduced-sized DispNet (32) (Small Enc-Dec), that has a number of parameters similar to our model (including both the global and the local module). Finally, we compare against Struct2Depth (34), current state-of-the-art system for unsupervised depth estimation.

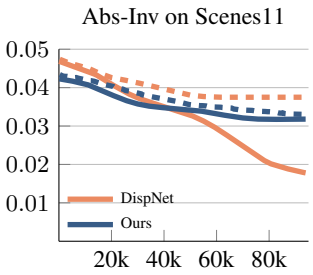

Figure 2: For large networks, the loss on training points (solid lines) is significantly higher than the validation loss (dashed lines). In contrast, our global-local architecture learns generalizible representations.

As shown in Table 1, our approach outperforms all the baselines in the sparse supervision regime. Specifically, we outperform the architecture of Eigen et al. (20) on average by 53%, the architecture of Laina et al. (33) by 22.5%, and the fully convolutional DispNet by 20%. Indeed, due to over-parametrization, these baselines tend to overfit to the training points, failing to generalize to unobserved images and locations. This is empirically demonstrated in Fig. 2, where we plot the depth loss on training points as a function of the number of iterations. Decreasing the size of the

Method	Scenes11			SUN3D			RGB-D		
	Abs-Inv	Abs-Rel	S-RMSE	Abs-Inv	Abs-Rel	S-RMSE	Abs-Inv	Abs-Rel	S-RMSE
Eigen (20)	0.045	0.57	0.77	0.072	0.82	0.38	0.046	0.54	0.37
DispNet (32)	0.038	0.51	0.70	0.041	0.49	0.33	0.038	0.45	0.36
FCRN (33)	0.041	0.52	0.74	0.047	0.44	0.30	0.042	0.45	0.35
Small Enc-Dec	0.046	0.66	0.83	0.064	0.73	0.45	0.049	0.58	0.46
Struct2Depth (34)	0.058	0.95	0.81	0.037	0.44	0.27	0.037	0.44	0.48
Ours	**0.031**	**0.43**	**0.61**	**0.035**	**0.37**	**0.25**	**0.033**	**0.37**	**0.33**

Table 1: In the sparse training regime, our method can efficiently learn to predict depth from single point supervision, outperforming significantly both standard architectures and unsupervised depth estimation systems.

Method	Scenes11		SUN3D		RGB-D	
	rot	trans	rot	trans	rot	trans
Scratch-MLP	1.3	74.4	3.6	55.5	5.3	78.4
Pretrained-MLP	0.9	26.7	2.7	32.5	4.4	51.5
Scratch-Full	**0.7**	10.3	1.8	25.0	**3.2**	30.5
Pretrained-Full	**0.7**	**9.2**	**1.7**	**22.4**	**3.2**	**28.7**
KLT Matlab (22)	0.9	14.6	5.9	32.3	12.8	49.6
8-point FlowFields (22)	1.3	19.4	3.7	33.3	4.7	46.1

Table 2: Estimation of camera motion based on the global parameters estimated by our model. We initialize the global module either randomly (Scratch) or as trained with our approach (Pretrained). We then append a small MLP and train supervised camera motion prediction by tuning either just the MLP (MLP) or the full network (Full). We report rotation (rot) and translation (trans) errors in degrees (since translation is normalized to 1).

architecture to address overfitting does not however solve the problem: the Small Enc-Dec, with number of parameters similar to our network, achieves poor results, mainly due to its limited capacity.

Our approach also achieves on average 24% better error than the unsupervised depth estimation baseline (34) over all datasets and metrics. Indeed, the considered datasets represent a challenge for geometry-based methods given the presence of large homogeneous regions, occlusions, and small baselines between views, which are typical factors encountered in indoor scenes. Noticeably, the performance of Struct2Depth on the SUN3D dataset is relatively good, boosted by the larger baseline between views and the abundance of features.

4.2 Global parameters and the camera motion

According to the intuition behind our model, the global parameters should have information about the observer's ego-motion between the frames, and as such should be related to the actual metric camera motion. Here we study this relation empirically, by training a camera pose predictor on the output of our global module, in supervised fashion. Note that this is done for analysis purposes only, after our full model has been trained: at training time the model has no access to the ground truth camera poses. Specifically, we add a small two-layer MLP with 256 hidden units on top of the global module that is either pre-trained with our method or randomly initialized. We then either train the full network or only the appended small MLP to predict the camera motion in supervised fashion (details of the training process are provided in the supplement). Results in Table 2 show that the global parameters indeed contain information about the camera pose. In both training setups pre-trained network substantially outperforms the random initialization: 17% to 64% error reduction across datasets and metrics when only tuning the MLP and up to 11% error reduction when training the full system. Interestingly, our method is also competitive against classic state-of-the-art baselines for motion estimation (22).

5 Conclusion

Motivated by the way natural agents learn to predict depth, we propose an approach for training a dense depth estimator from two unconstrained images given only very sparse supervision at training time and without the explicit use of geometry. We show that in cluttered indoor environments our global-local model outperforms state-of-the-art architectures for depth estimation by up to 20% in the sparse data regime.

REFERENCES

[1] R. Held, Y. Ostrovsky, B. de Gelder, T. Gandhi, S. Ganesh, U. Mathur, and P. Sinha, "The newly sighted fail to match seen with felt," *Nature neuroscience*, vol. 14, no. 5, p. 551, 2011.

[2] I. P. Howard and B. J. Rogers, *Seeing in depth, Vol. 2: Depth perception.* University of Toronto Press, 2002.

[3] E. B. Goldstein and J. Brockmole, *Sensation and perception*. Cengage Learning, 2016.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, second Edition.

[5] R. Garg, B. G. V. Kumar, G. Carneiro, and I. D. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 740–756.

[6] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 6612–6619.

[7] H. Wallach and D. O'Connell, "The kinetic depth effect." *Journal of experimental psychology*, vol. 45, no. 4, p. 205, 1953.

[8] G. Johansson, "Visual motion perception," *Scientific American*, vol. 232, no. 6, pp. 76–89, 1975.

[9] K. Nakayama and J. Loomis, "Optical velocity patterns, velocity-sensitive neurons, and space perception: a hypothesis," *Perception*, vol. 3, no. 1, pp. 63–80, 1974.

[10] J. J. Koenderink, "Optic flow," *Vision research*, vol. 26, no. 1, pp. 161–179, 1986.

[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 2758–2766.

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17

[13] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[14] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *ECCV 2016 Workshops*, 2016.

[15] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *AAAI Conf. Artificial Intell.*, 2018.

[16] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, no. 293, pp. 133–135, 1981.

[17] R. I. Hartley and P. F. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[18] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 6, pp. 580–593, 1997.

[19] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883. Springer Verlag, 2000, pp. 298–372.

[20] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 2366–2374.

[21] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *arXiv:1907.01341*, 2019.

[22] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 5038–5047.

[23] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[24] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 322–337.

[25] D. J. Rezende, S. M. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3d structure from images," in *Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 4997–5005.

[26] S. M. A. Eslami, D. Jimenez Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis, "Neural scene representation and rendering," *Science*, vol. 360, no. 6394, pp. 1204–1210, 2018.

[27] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Trans. Robot.*, vol. 24, no. 5, 2008.

[28] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2320–2327.

[29] J. Xiao, A. Owens, and A. Torralba, "Sun3D: A database of big spaces reconstructed using sfm and object labels," in *Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 1625–1632.

[30] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2012, pp. 573–580.

[31] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[32] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[33] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.

[34] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[35] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 6602–6611.

6 APPENDIX

6.1 CONNECTION WITH TWO-VIEW TRIANGULATION

The problem of triangulation consists of computing the 3D coordinates of a point given its (noisy) projections on two or more views and the camera parameters of the views. Hence, it is a geometric problem. Following the usual formalism of homogeneous coordinates (4), the perspective projection of a 3D point M on two cameras with projection matrices P_1, P_2 (comprising the intrinsic and extrinsic parameters of both views) is given by $\lambda_1 m_1 = P_1 M$ and $\lambda_2 m_2 = P_2 M$, where λ_1, λ_2 are the projective scaling factors.

Given P_1, P_2, m_1, m_2, the linear triangulation algorithm (4, Sec.12.2), which tackles the triangulation problem in its most general setting (projective cameras), computes the 3D point M by minimizing the Rayleigh quotient $\|AM\|/\|M\|$, where A is the matrix

$$A(P_1, P_2, m_1, m_2) = \begin{pmatrix} [m_1]_\times P_1 \\ [m_2]_\times P_2 \end{pmatrix} \tag{1}$$

and $[u]_\times$ is the cross product matrix (such that $[u]_\times v = u \times v$, for all v).

In case of multiple point correspondences $\{m_1^i \leftrightarrow m_2^i\}$ for $i = 1, \ldots, N$, the camera matrices P_1, P_2 appear in the triangulation equations (1) of all of them, and hence, are "global" variables. If the camera matrices are known, then (i) every 3D point M^i can be triangulated independently from the rest, and (ii) the triangulated point is a function of the point correspondences $m_1^i \leftrightarrow m_2^i$.

$$M^i = f(m_1^i, m_2^i; P_1, P_2). \tag{2}$$

This intuition inspired the design of our modular architecture: First, a neural network regresses global variables which depend only on the two views, and then those global variables are used by a local module to generate a fully-convolutional net which transforms point correspondences (optical flow) into depth.

6.2 TRAINING PROCESS

We train our model from scratch on the Scenes11 dataset for approximately 150K steps using Adam as optimizer with an initial learning rate of $1e-4$. We normalize all losses with the number of points used to compute them. The loss weights for depth and smoothness λ_p, λ_s are 5.0 and 2.0 respectively. To increase generalization, we perform data augmentation at training time by mirroring pairs on the x-axis and rotating them 180 degrees, both 50% probability. For a fair comparison, we trained all baselines with exactly the same strategy and hyper-parameters on a desktop PC equipped with an NVIDIA-GeForce 940MX.

For the pose experiments in Sec. 4.4, we trained a 2 hidden layer MLP with 20 nodes and leaky ReLU activation function to predict relative camera motion between frames from the global parameters estimated by our global network. For all datasets and all variations, we trained with an L_1 loss between estimated and real camera poses for 50K steps.

To encourage the local smoothness of the predicted depth maps, we add an L_1 regularization penalty on the gradient $\nabla\hat{\mathbf{z}} = (\partial_x\hat{\mathbf{z}}, \partial_y\hat{\mathbf{z}})$ of the estimated inverse depth. Similarly to classic structure from motion methods and unsupervised depth learning literature (35), we modulate this penalty according to the image gradients $\partial\mathbf{I}_1$, allowing depth discontinuities to be larger at points with large $\partial\mathbf{I}_1$.

6.2.1 TUNING OF BASELINES

To fairly study the sparse training setting, we tuned the baseline to reach the best performance on the sparse training task. First, we changed the input layer of each baseline architecture. Exactly as for ours, the baselines' input consists of the concatenation of the image pair and the optical flow on the last channel. Given the very sparse supervision signal, we noticed that the ReLu activation function generated extremely sparse and noisy gradients. Therefore, we modified the original activation function of DispNet (32), FCRN (33) and Eigen (20) from ReLu to LeakyRelu. This change improved the performance of the baselines of up to 50% on average over metrics.