Adapting Depth Prediction Networks to Unknown Domains Using Pose Information

Matthias Brucker
ETH Zurich
mbrucker@ethz.ch

Davide Plozza
ETH Zurich
dplozza@ethz.ch

Tomasz Zaluska
ETH Zurich
zaluskat@ethz.ch

Hamza Javed
ETH Zurich
javedh@ethz.ch

Jeremaine Siegenthaler
ETH Zurich
jsiegent@ethz.ch

Abstract

Depth prediction networks have been successfully implemented in dense monocular SLAM pipelines, but such networks often suffer from drops in accuracy in unseen domains. This paper provides a proof of concept of adapting pretrained networks to unseen domains by leveraging pose information. We explore the use of two different kinds of poses, namely ground truth poses and poses obtained from a parallel working SLAM system. This allows an end-to-end optimisation of the whole pipeline. Our results show successful adaption to an unseen domain in multiple scenes that are not possible without the additional pose information. The work can be seen as a first step towards an adaptive end-to-end optimizable dense monocular SLAM pipeline.

1. Introduction

Dense monocular simultaneous localization and mapping (SLAM) is a highly active research area and has a lot of potential applications in robotics and general navigation tasks. Recent works have combined monocular depth prediction networks with SLAM systems [10] [11], only requiring RGB images as input. However, the drop in accuracy of monocular depth prediction networks that are caused by domain-shifts limits the success of such systems.

Monocular depth prediction networks have become the state of the art for monocular depth prediction and match the performance of their stereo counterparts. Recent works have also shown the remarkably successful use of unsupervised training methods which do not require expensive and often unavailable depth sensor information [12] [6]. The photometric losses used by such systems however often rely on brightness consistency assumptions which limits their baseline to a few consecutive frames.

It was proposed to tackle the problem of domain shifts by implementing online learning for depth prediction networks [2]. Since by definition no external supervision signal is available at deploy time, self-supervised training methods must be used. One possibility to minimize the limitations of those could be the addition of other information, that can be obtained by sensors that are often already implemented in robotic systems. Pose estimations (especially relative poses) can be obtained by deploying a traditional monocular pose estimator or from inertial measurement units (IMU) that are commonly integrated in robotic platforms.

We want to explore the concept of integrating pose information in a self-supervised depth prediction network to adapt a pretrained network to an unseen environment.

2. Methods and Architecture

2.1. Datasets

Multiple RGB datasets exists that can be used to train and validate monocular depth prediction networks, most notably KITTI [5] and Cityscapes [3] for outdoor applications and NYU-v2 [8], ScanNet [4] and TUM [9] for indoor applications. To keep the domain shift of our experiments within reasonable boundaries and since monocular depth predictors are often used for dense indoor SLAM, we decided to stay within the indoor domain. In particular, we focus on bridging the domain gap between NYU-v2 (which was originally used to train the depth prediction network) to previously unseen TUM sequences. TUM is especially convenient, since it provides both ground truth (GT) poses and depth (for validation purposes).

2.2. Depth Prediction Network

Since complete training of a depth prediction network is outside the scope of this project, we need a pre-trained

depth prediction network. It must be state of the art, self-supervised, and preferably have source code and pre-trained weights publicly available. These criteria lead us to the SC-SfM Learner [1] network, which is state of the art for self-supervised depth prediction on the NYU-v2 dataset. Since poses are required for the self-supervised reprojection losses, SC-SfM Learner contains a pose prediction module. In this project, we replace this module GT or SLAM poses instead.

2.3. Loss Function

Following [1] the overall loss function is defined as

$$L = \alpha L_p^M + \beta L_{GC} + \gamma L_s, \tag{1}$$

where α, β and γ constitute the weighting parameters of their corresponding loss function. L_p^M is a photometric loss and is defined as follows:

$$L_p^M = \frac{1}{|V|} \sum_{p \in V} (\lambda_i ||I_a(p) - I_{a'}(p)|| + \lambda_s \frac{1 - SSIM_{aa'}(p)}{2})$$

An image dissimilarity loss SSIM is used to handle illumination changes. V stands for the set of valid points p that are successfully reprojected from I_b into the image plane of I_a. The reprojection is denoted as $I_{a'}$. In the original paper of Bian et. al. [1], it is a function the predicted depth map D_a, the predicted pose P_{ab}, and image I_b.

$$I_{a',bian2020} = func(P_{ab}, D, I_b) \tag{2}$$

Our approach uses a slightly alternate form, with the difference that instead of taking the pose from the pose prediction network, we take either the ground truth relative pose $P_{ab,gt}$ or the relative pose generated by SLAM $P_{ab,slam}$:

$$I_{a',ours} = func(P_{ab,gt}, D_a, I_b) \tag{3}$$

$$I_{a',ours} = func(P_{ab,slam}, D_a, I_b) \tag{4}$$

L_{GC} is the geometric consistency loss and depends on a computed depth inconsistency map D_{diff}:

$$L_{GC} = \frac{1}{|V|} \sum_{p \in V} D_{diff}(p) \tag{5}$$

Furthermore, a smoothness loss L_s is mainly used for regularization and is defined as:

$$L_s = \sum_p (e^{-\nabla I_a(p)} \cdot \nabla D_a(p))^2 \tag{6}$$

2.4. GradSLAM

For the SLAM module in our pipeline, we decided to use the simple Iterative Closest Point (ICP) SLAM implementation from the GradSLAM library. GradSLAM is a fully differentiable dense SLAM system [7]. The GradSLAM module takes a sequence of images and predicted depths as input and provides differentiable visual odometry which we use as pose information. Due to its differentiability, we can flow gradients through the SLAM module in order to optimize our depth prediction network.

3. Experimental Setup

Given the choice of methods described above, this project aims at answering the following research question: To what extent can pose information be used to adapt the SC-SfM Learner depth prediction network trained on NYU-v2 to unseen TUM sequences?

3.1. Pose Prediction Network (SC-SfM Learner)

As a baseline we perform adaptation by continuing the training from SC-SfM Learner (depth prediction network and pose prediction network, same hyperparameters). We initialize both networks with weights pretrained on NYU-v2.

3.2. GT Poses

As a first step, we show that depth prediction on TUM sequences can be improved by using GT poses. For this purpose, we compute the reprojection using Eq. 3 and optimize the loss function from Eq. 1. The high level system architecture is depicted in Figure 1.

3.3. SLAM Poses

In a second step, we assume that no GT poses are available, and explore whether poses generated through ICP SLAM from predicted depth maps can improve depth prediction on TUM sequences (see Figure 2). Our hypothesis is that by using ICP instead of the pose network, adaptation learning will lead to depth predictions that are more consistent for ICP, and thus be optimized for 3d reconstruction via dense SLAM.

Here, the loss function from Eq. 1 is realized using the reprojection of Eq. 4. The relative poses are generated by performing ICP SLAM on the two input frames a and b. We experiment with two scenarios for optimizing the loss:

1. The gradients flow directly from the loss function to the SC-SfM Learner module trough the predicted depths and not through ICP SLAM.

2. The gradients flow through GradSLAM poses and the SC-SfM Learner module as well. This is only possible since GradSLAM is differentiable.

Figure 1: System architecture with GT poses.

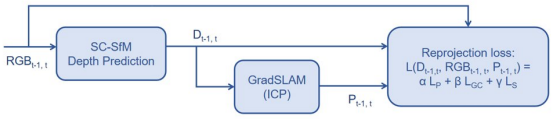

Figure 2: System architecture with poses from SLAM.

To evaluate the performance of the ICP SLAM poses, we implemented a MSE loss between the GT poses and the SLAM poses (we decompose the poses into a translational vector and rotational euler angles vector and compute the MSE loss for each). We do not use this loss for training.

3.4. Sequences and Training Setup

We use four sequences of the TUM RGB-D SLAM dataset for our experiments. The arguably simplest sequence *freiburg2_xyz* covers a small volume of an office desk with only small pose differences between frames and almost no rotation. *freiburg1_rpy* shows a similar scene but the extensive camera motion is almost pure rotation. *freiburg1_desk* covers a bigger volume than the two previous sequences and combines translational with rotational camera motion. The *freiburg_pioneer_slam* sequence contains only translational and rotational motion in the x-y plane and is characterized by a big difference in fore- and background depth.

Our experiments are conducted with a batch of two sequences of ten frames each to simulate the limited data available in an online refinement process. (Exception with SLAM poses: to simplify gradient flow, we use ten sequences of two frames.) We overfit the SC-SfM Learner depth prediction network on these 20 images in 300 iterations. For validation, we compare the predicted depth map to GT depth images provided by the TUM dataset. In order to ensure consistent scale in the predicted depth map, median scaling is used before training and during validation. The source code to reproduce the experiments is provided on Github.[1]

[1] https://github.com/aquamin9/End-2-end-self-supervised-SLAM

4. Results

4.1. Results from Pose Prediction Petwork (SC-SfM Learner)

From the validation results in Table 1, it can be seen that adaptation with poses form SC-SfM Learner's pose prediction network failed in *freiburg2_xyz* and *freiburg2_pioneer_slam*. A slight depth improvement could be observed for *freiburg1_desk*, but the model was still outperformed by adaptation with GT poses as well as with SLAM poses.

4.2. Results from GT Poses

Our experiments with GT poses show a decreasing validation error for three sequences *freiburg2_xyz*, *freiburg1_desk* and *freiburg2_pioneer_slam*, see Figure 4. The relevant photometric and geometric losses consistently decrease as training progresses, only the regularizing smoothness loss increases towards the end, as the network learns the details of the sequences. The validation metrics in Table 1 show that depth predictions after adaptation with GT poses consistently outperform both the original depth predictions as well as the ones after adaptation with the pose network. The quantitative results are backed up by considerably improved qualitative depth predictions as shown in Figure 3.

On the *freiburg1_rpy* sequence, we could not achieve improved depth prediction. Instead, the predicted depth smoothens out as training progresses. The reason is too little translational pose difference between frames, as shown in Figure 5 and discussed in section 5.1. This sequence was therefore ommited in our validation results (Table 1).

4.3. Results from SLAM Poses

As expected, the results with SLAM poses are not as good as the results with GT poses, since ICP converges to an inexact pose when the depth prediction is not consistent enough.

1. When flowing the gradients only through SC-SfM Learner and not through the SLAM poses, adaptation failes. Due to inexact poses, the validation loss does not decrease and the model converges to a sub-optimal local minimum.

2. When flowing the gradients through SC-SfM Learner and GradSLAM poses at the same time, the validation loss decreases in *freiburg2_xyz* and *freiburg1_desk*. Furthermore, we observe that the MSE between the poses generated from ICP and the GT poses decreases (even though we do not optimize this loss).Although some metrics indicate performance of SLAM poses being on a par with GT poses, upon visual inspection

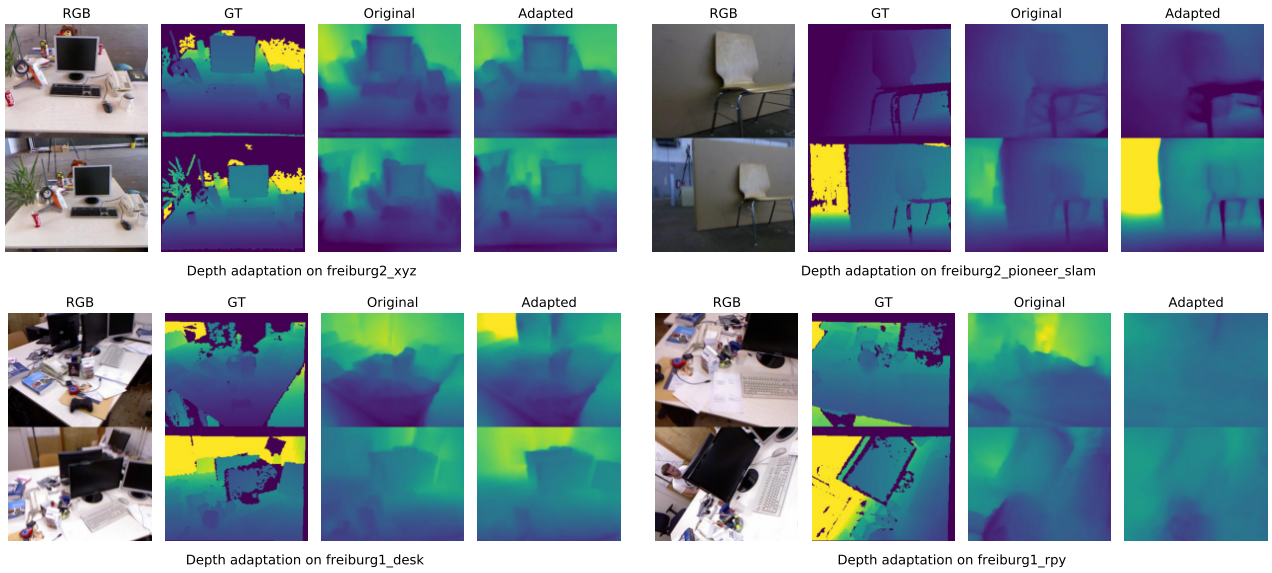

Figure 3: Qualitative results of the SC-SfM Learner depth prediction network before and after adaptation with GT poses on four different TUM sequences. For the three datasets *freiburg2_xyz* (better desk details in upper left part of image), *freiburg2_pioneer_slam* (better background foreground contrast) and *freiburg1_desk* (better representation of screens), considerable improvements in depth prediction could be observed. On *freiburg1_rpy*, training lead to smoothing due to the small translational movement between camera frames.

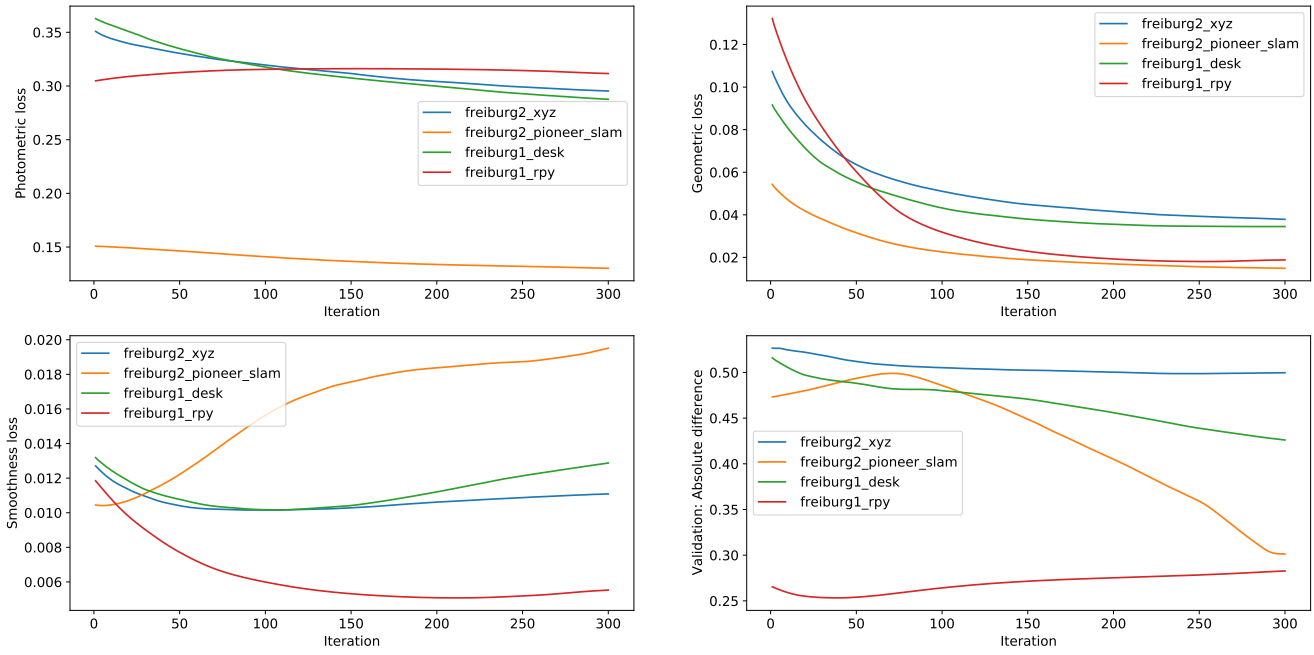

Figure 4: Overview of the loss functions and the validation error of all four used datasets. The absolute difference validation error (bottom right) decreased for all except *freiburg1_rpy*. The geometric loss (top right) decreased the most. The photometric loss (top left) decreased only for the datasets that show an improved validation error. The regularizing smoothness loss (bottom left) increases towards the end except for *freiburg1_rpy*.

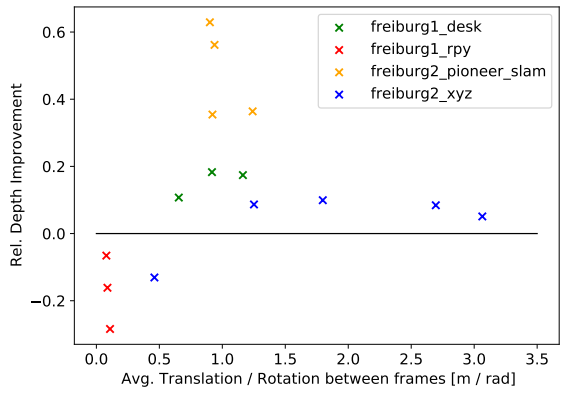

Figure 5: Relative depth improvements vs. the translation to rotation ratio for different sequences with frame skipping.

(see Figure 6), the depth maps resulting from adaptation with SLAM poses are generally worse than those from GT poses. SLAM poses do not yield depth improvement in *freiburg2_pioneer_slam*.

5. Discussion

5.1. GT Poses

The qualitative and quantitative results of our experiments with GT poses show that the use of GT poses significantly improves depth prediction on unknown TUM sequences, whereas we could not achieve any improvement with the original pose network architecture. This supports our hypothesis that accurate pose information is indeed necessary to enable fast adaptation to unseen domains. However, there are some limitations related to the choice of the sequences.

- We could observe that the adaptation on the depth network failed on the *freiburg1_rpy* sequence, a sequence with very small translation motion and very strong rotational motion between frames. Therefore, we use frame skipping to evaluate the influence of pose difference between frames on training performance. The results in Figure 5 show no general correlation between pose difference between frames and training success. It seems like other characteristics inherent to the selected sequence and frame have a stronger effect. However, if translations between frames are too small (below a certain threshold of translation over rotation ratio), the depth prediction cannot improve since reprojection becomes independent of depth. As a result, learning fails on *freiburg1_rpy* and the loss is optimized through smoothing of the depth image.
- Another limitation of depth prediction are untextured regions, where reprojection can be somewhat ambiguous. In Figure 3, this problematic can be observed on the computer screens, where the depth of its border seems accurate, whereas the predicted depth of its center would yield a curved 3d reconstruction.

5.2. SLAM Poses

As described in section 4.3, adaptation with SLAM poses was not possible in the first scenario, where no gradients were flown through the poses. The reasoning for that is that since the depth predictions are bad, ICP fails to find meaningful poses, which then leads to unsuccessful training.

In the second scenario, when flowing the gradients through GradSLAM, we observed that both the training losses and the validation loss decreased for two of our three sequences. Even though some validation metrics suggest otherwise, qualitative analysis shows that depth predictions after adaptation with SLAM poses are consistently worse than with GT poses. In particular, the depth images loose contrast, as the depth network is optimized to provide depth maps that allow ICP to compute good poses for reprojection. Indeed, we observed that the poses obtained via ICP converge to the GT poses. In addition to the limitations of adaptation with GT poses, SLAM poses come with new unknowns which are highly sequence dependent and cause adaptation to fail at times, as our results on *freiburg2_pioneer_slam* show. However, since adaptation with SLAM poses beat the pose network baseline in two sequences (Table 1), we could show that end-to-end adaptation of a depth prediction network with SLAM is possible. For a more rigorous statement on the feasibility of adaptation with SLAM poses, more experiments need to be conducted.

6. Hyper-Parameter Selection

The selection of hyper-parameters that control the online training is an important factor for the experiment. For the main part, we use the hyper-parameters as suggested by [1]. Our experiments have shown that only single scale reprojection without using down-sampled depth maps yields satisfactory results, which can be traced back to the bad quality of the depth predictions on the unseen domain.

Loss Weights We experimented with different weighting parameters α, β and γ for the losses. The photometric reprojection loss L_p^M is essential for training and usually weighted highest. Training fails when it is not used (Figure 7). The use of a combination of photometric and geometric loss L_{GC} resulted in the highest error improvements, we use $\alpha = 1$ and $\beta = 0.5$ in our experiments. The smoothness loss L_s is only used for normalization and set to $\gamma = 0.1$ for all experiments.

Sequence	Architecture	Abs Diff	Abs Rel	Sq Rel	a1	a2	a3
freiburg2_xyz	Original	0.527	0.383	0.517	0.472	0.614	0.696
freiburg2_xyz	Posenet Poses	0.515	0.394	0.530	0.456	0.620	0.696
freiburg2_xyz	GT Poses	0.500	**0.337**	0.505	**0.562**	0.671	**0.713**
freiburg2_xyz	SLAM Poses	**0.488**	0.356	**0.484**	0.508	**0.680**	0.703
freiburg2_pioneer_slam	Original	0.473	0.263	0.679	0.698	0.765	0.784
freiburg2_pioneer_slam	Posenet Poses	0.578	0.330	1.006	0.696	0.753	0.777
freiburg2_pioneer_slam	GT Poses	**0.301**	**0.142**	**0.137**	**0.772**	**0.848**	**0.862**
freiburg2_pioneer_slam	SLAM Poses	0.494	0.329	0.369	0.471	0.683	0.750
freiburg1_desk	Original	0.516	0.336	0.384	0.448	0.676	0.750
freiburg1_desk	Posenet Poses	0.461	0.294	0.343	0.552	0.703	0.757
freiburg1_desk	GT Poses	**0.426**	**0.256**	**0.290**	0.611	0.738	0.779
freiburg1_desk	SLAM Poses	0.495	0.279	0.773	**0.697**	**0.762**	**0.783**

Table 1: Quantitative validation results of the SC-SfM Learner depth prediction network for *freiburg2_xyz*, *freiburg2_pioneer_slam* and *freiburg1_desk* before (original) and after adaptation with poses from the SC-SfM Learner pose network, GT poses and SLAM poses. Metrics are taken from [1]. For the first three error metrics, lower is better, for the last three accuracy metrics, higher is better. Best results in a sequence are marked in bold.

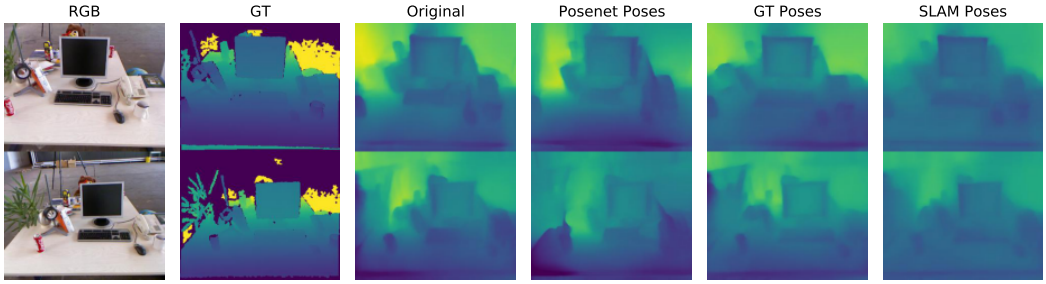

Figure 6: Qualitative results of the SC-SfM Learner depth prediction network for *freiburg2_xyz* before (original) and after adaptation with poses from the SC-SfM Learner pose network, GT poses and SLAM poses.

Optimization In order to reduce training time and only finetune the model, we test out the training with and without freezing the encoder part of the network. Our experiments with optimization parameters (Figure 8) show that freezing the encoder part of the model yields worse results, so we do not use freezing in our experiments. Increasing the initial learning rate from 10^{-6} to 10^{-5} with Adam optimization leads to faster convergence at comparable validation accuracy (20 iterations, ca. three minutes on GPU), which is a hint that further parameter tuning can significantly improve adaptation performance. During our main experiments, we keep the learning rate at 10^{-6} as suggested by [1].

7. Conclusion and Future Work

Our work shows the potential of using pose information to adapt a depth prediction network to a new environment using only 20 frames. In particular, we present a proof of concept that GT and SLAM pose information can be used to adapt the SC-SfM Learner depth prediction network (trained on NYU-v2) to TUM sequences with some limitations. This can be seen as a first step towards an adaptive end-to-end optimizable dense monocular SLAM pipeline. Future work should try to implement our findings in a truly online fashion. This requires extensive hyperparameter tuning and architectural refinements. Our loss functions could also be combined with other SLAM methods, e.g. voxel- instead of pointcloud-based SLAM for direct reprojection of the 3D SLAM map.

8. Work Distribution

The work was equally distributed among all team members.

References

[1] J.-W. Bian, H. Zhan, N. Wang, T.-J. Chin, C. Shen, and I. Reid. Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. 2020.

[2] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *CoRR*, abs/1811.06152, 2018.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[6] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CoRR*, abs/1609.03677, 2016.

[7] K. M. Jatavallabhula, G. Iyer, and L. Paull. grad-slam: Dense slam meets automatic differentiation. *CoRR*, abs/1910.10672, 2019.

[8] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[9] 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*, pages 573–580, 2012.

[10] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. *CoRR*, abs/1704.03489, 2017.

[11] L. Tiwari, P. Ji, Q.-H. Tran, B. Zhuang, S. Anand, and M. Chandraker. Pseudo rgb-d for self-improving monocular slam and depth prediction. *ECCV 2020*, 2020.

[12] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *CoRR*, abs/1704.07813, 2017.

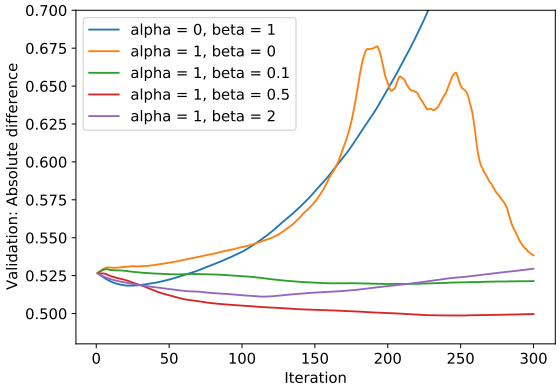

Figure 7: Ablation study on loss parameter weights on *freiburg2_xyz* ($\gamma = 0.1$ always).

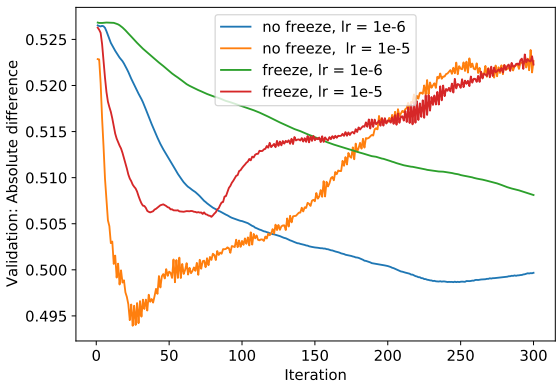

Figure 8: Ablation study on learning rate and encoder freezing on *freiburg2_xyz*.