# Adaptive Learning for Hybrid Visual Odometry

Ziming Liu, Ezio Malis, Philippe Martinet

# Adaptive Learning for Hybrid Visual Odometry

Ziming Liu, *Student Member, IEEE,* Ezio Malis, *Member, IEEE,* and Philippe Martinet, *Member, IEEE*

***Abstract*—Hybrid visual odometry methods achieve state-of-the-art performance by fusing both data-based deep learning networks and model-based localization approaches. However, these methods also suffer from deep learning domain gap problems, which leads to an accuracy drop of the hybrid visual odometry approach when new type of data is considered. This paper is the first to explore a practical solution to this problem. Indeed, the deep learning network in the hybrid visual odometry predicts the stereo disparity with fixed searching space. However, the disparity distribution is unbalanced in stereo images acquired in different environments. We propose an adaptive network structure to overcome this problem. Secondly, the model-based localization module has a robust performance by online optimizing the camera pose in test data, which motivates us to introduce test-time training machine learning method for improving the data-based part of the hybrid visual odometry model.**

***Index Terms*—Deep Learning for Visual Perception, Visual Learning, Computer Vision for Transportation**

## I. Introduction & background

AMONG the many algorithms for visual odometry, hybrid visual odometry method has proven to be a successful solution for localization [1], [2], [3], [4]. For these hybrid visual odometry methods, there are three main approaches: dense direct odometry [3], sparse feature-based odometry [2] and sparse direct odometry [4], [5]. Hybrid visual odometry combines a deep neural network that predicts the depth map of the environment, and a model-based pose estimation module that computes the camera pose [3]. Usually, in hybrid visual odometry, the model-based localization module can realize robust performance by online optimization on test data. However, the data-based part suffers from the domain gap problem on different test datasets. To fix the domain gap and achieve a better result on a new test dataset, the data-based model usually requires to be trained again. This introduces more training costs and labor costs for data annotations.

Depth estimation networks in hybrid visual odometry can be optimized with supervised or unsupervised methods [6], [7], [8], [3]. These supervised networks are trained with L1 or L2 photometric loss on the ground truth depth maps. From the network structure, they can be grouped into monocular and stereo networks. Firstly, the monocular network is a typical regression model. various state-of-the-art foundation networks are explored for monocular depth estimation, including convolution networks, transformer networks [9], [10], [11]. Besides the network structure, the network initialization also performs a critical role for a monocular network [12], because monocular networks are learning knowledge from the data. In contrast, stereo networks are learning stereo matching relations, which is more reliable for different data domains. Firstly, 2D CNN networks, e.g. DispNet [13], are used to regress disparity similar to monocular networks [8]. Secondly, two-stage 2D-3D CNN networks have more robust and accurate performance [6], [14]. GCNet [14] proposes the important soft argmin method for disparity regression. It is widely used in the following works [6], [9]. PSMNet [6] further modified 2D encoder and 3D matching netwroks with pyramid convolution structure and cascade disparity regression. CascadeStereo [15] predicts disparity map from coarse to fine with cascaded PSMNet [6] or GWCNet [16]. The next disparity searching space is initialized by the last disparity prediction. Thirdly, recurrent stereo networks [17], [18], [7] achieve better accuracy, especially on the high-resolution image, but taking more computation costs. RAFTStereo [17] first builds a recurrent network based on RAFT optical flow network. Then CREStereo [18] improve it to a cascaded recurrent network. IGEVStereo [7] combines the typical two-stage stereo network with the recurrent stereo network. Although the recurrent stereo network has high accuracy, the computation cost is much higher. Overall, two-stage stereo network has a better balance on accuracy and cost.

Currently, stereo networks build the stereo cost volume and compute disparity regression in a pre-defined disparity searching space [6], [15]. Although these stereo networks have shown impressive performance, they suffer from a common problem: the disparity distributions are different when acquiring stereo images in different environments. The unbalanced disparity distribution in test datasets has been discussed [19]. This problem makes the pre-defined disparity searching space hard to learn the disparity fast and well. Some previous works [15], [9] have shown the advantage of adaptive disparity distribution center. To be efficient, a monocular sub-network is introduced to predict the initialized disparity distribution center for the stereo matching network.

Furthermore, the domain gap between training and test data also affects the performance of data-based model. The domain gap can be solved by re-training on the target dataset or unsupervised domain adaption method which is a lower cost solution. Motivated by the model-based localization module in hybrid visual odometry, which predicts the camera pose with online optimization. A new unsupervised domain adaption method, visual odometry test-time training (TTT), is proposed. It is an online optimization method on the test dataset. Test-Time Training has been explored on computer vision tasks [20], [21], [22], [23], [24], [25]. Previous methods can be grouped into two types. Firstly, the networks with main task and auxiliary self-supervised task are jointly optimized. For example, the image rotation prediction auxiliary task is jointly trained with the image classification task in [20]. A contrastive learning task [21], image reconstruction by masked autoencoder [22] are used as an auxiliary self-supervised task of test-time training. Secondly, Some test-time training methods do not change the training loss function, instead of using regularization methods in the testing time. For example,

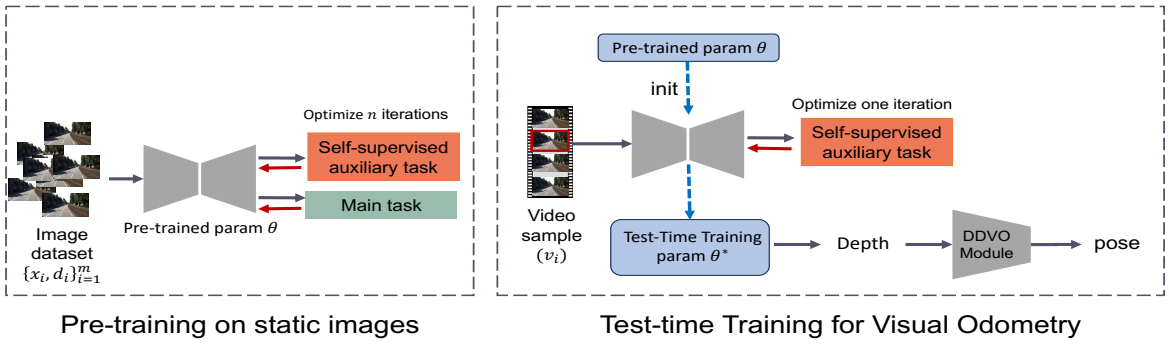



Fig. 1. **Left:** Training of the pre-trained depths. At this stage, the stereo depth network is trained with both supervised main task and self-supervised auxiliary task on the image & disparity pairs $(x_i, d_i)$. **Right:** Process of Test-Time Training for visual odometry. At this stage, there are three steps, firstly, the network is initialized with pre-trained parameters, then, the self-supervised auxiliary task updates the network for one iteration to obtain new network parameters $\theta^*$. Finally, the TTT network $\theta^*$ is composed of the traditional dense direct visual odometry module to output camera pose $\hat{p}_i$.

TENT [25] minimizes the entropy of output distribution at the test stage. MEMO [23] minimizes the entropy of different augmentations. SHOT [24] proposes information maximization regulation for source-free adaption. For the first kind of test-time training, self-supervised task is the most important part.

Among depth estimation tasks, self-supervised methods have already been explored well. The self-supervised depth estimation is based on two types of loss functions, including the self-supervised stereo matching loss (L1 loss between warped stereo image and reference image [26]) and temporal matching loss (L1 loss between warped temporal image and reference image [8]). Monodepth [26] is one of the earliest self-supervised depth estimation methods with stereo matching loss. Similarly, SfmLearner [27] realizes the self-supervised training with temporal matching loss. More recently, the combination methods of stereo matching loss and temporal warping loss are widely used [8].

The main contributions are shown as follows:

- A new disparity-adaptive stereo network with a learnable monocular initialization branch is proposed. It aims to solve the disparity domain gap of different images.
- Test-time training for hybrid visual odometry is proposed. It aims to solve the domain gap between training and test data.

This paper is organized as follows. Section I describes the motivation, background and contributions. Section II shows the proposed method. The experiment results are shown in Section III. Finally, Section IV concludes this paper.

## II. The Proposed Approach

This paper introduces a novel approach called adaptive hybrid visual odometry which is based on the hybrid visual odometry [3]. There are two adaptive aspects: adaptive disparity prediction with a new network structure and adaptive optimization with test-time training.

Firstly, to address the disparity distribution imbalance problem in different data domains, a new depth network is proposed, outlined in Section II-A, which is pre-trained on a still image dataset using both supervised depth estimation and self-supervised stereo matching tasks, as illustrated in Fig. 1 (Left).

Secondly, domain adaption for localization is explored on the visual odometry dataset. Visual odometry test-time training with single-time online optimization is used to replace the re-training on the target dataset. Test-Time Training strategy can generalize the pre-trained hybrid visual odometry model to a new data domain in an unsupervised way, as depicted in Fig. 1 (Right). Additional details are provided in Section II-B.

### A. Stereo network with adaptive disparity

This section presents the proposed stereo network with adaptive disparity searching space.

*1) Baseline stereo network:* The new network is built on a two-stage deep stereo matching framework [6], [28]. Two-stage stereo networks are composed of four modules: feature extraction network, 4D stereo cost volume, stereo matching network, and disparity regression layer. Feature extraction network is usually a shallow 2D convolution network that shares the same parameter for left and right images. Stereo cost volume is a 4D volume, with $Channel \times Disparity \times Height \times Width$ size. It is built by concatenating the left image and the shifted right image, which is generated by the pre-defined adaptive disparity searching space, across the feature channel. Stereo matching network is usually a 3D auto-encoder. Finally, a disparity regression layer predicts the logistic probability over a pre-defined adaptive disparity searching space. Disparity map is predicted by the weighted sum of the probability and the search space. The pre-defined disparity search with a disparity range of $0-192$ is commonly used in most public datasets [29], [30], [31].

*2) Adaptive disparity searching space:* In this paper, a new stereo-matching network is proposed. It generates an adaptive disparity searching space $\tilde{\mathbb{S}}$ computed by a monocular depth prediction branch which can predict disparity map $\hat{\mathbf{D}}$, as illustrated in Fig. 2.

Firstly, adaptive searching space $\tilde{\mathbb{S}}$ is obtained with predicted initial disparity map $\hat{\mathbf{D}}$ and bidirectional shifts $\pm R$, as shown in Eq. 1.

$$\tilde{\mathbb{S}} = [\hat{\mathbf{D}} - R, ..., \hat{\mathbf{D}} - 1, \hat{\mathbf{D}}, \hat{\mathbf{D}} + 1, ..., \hat{\mathbf{D}} + R] \tag{1}$$

where $\hat{\mathbf{D}}$ is the center of disparity distribution.

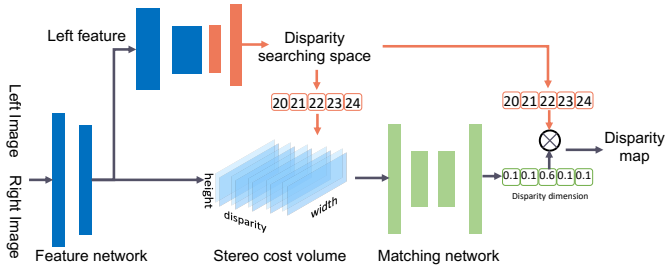



Fig. 2. Structure of stereo network with adaptive disparity. Disparity searching space is shown in disparity dimension. The values of disparity searching space are computed using a monocular predicted disparity map (e.g. $\hat{D}(\mathbf{p}) = 22$ ) and bidirectional shifts (e.g. $\pm$ 2 ). $\mathbf{p}$ is the pixel position.

Finally, values of $\tilde{\mathbb{S}}$ will be truncated into zero if they are negative, because disparity map should be positive in this task.

In contrast, pre-defined disparity searching space $\overline{\mathbb{S}}$ is generated as follows. It is fixed for any disparity distribution.

$$\overline{\mathbb{S}} = [\mathbf{D}_0, \mathbf{D}_0 + 1, ..., \mathbf{D}_0 + 2R], \mathbf{D}_0 = \mathbf{0} \tag{2}$$

With the disparity searching space, raw stereo cost volume is obtained by concatenating left feature maps and warped right feature maps. The warped right feature maps are computed by warping right feature maps using disparity searching space $\tilde{\mathbb{S}}$ which is a set of disparity maps. For the final disparity prediction, it is obtained by computing the weighted sum of $\tilde{\mathbb{S}}$ and weights $\in [0,1]$ learned by matching network. The weighted sum is computed in the disparity dimension.

Additionally, the feature extractor network in two-stage stereo network is replaced with general and lightweight encoder networks, such as ResNet [32] and MobileNetv2 [33]. New encoders are more efficient and effective.

### B. Visual odometry test-time training

Firstly, the basic test-time training method [34] is introduced. Then, basic test-time training is applied for hybrid visual odometry. Furthermore, a sequential test-time training visual odometry is proposed.

*1) Standard test-time training :* Test-Time Training is a machine learning method originally proposed for improving the accuracy of the image recognition problem. The standard Test-Time Training method has a self-supervised learning task to help the main task [34]. The network parameters $\theta = (\theta_1, .., \theta_K)$ of the $K$ layers can be divided into three groups: backbone network $\theta_b$, main classification head $\theta_m$, and self-supervised head $\theta_s$, respectively. By incorporating the self-supervised learning task, the objective function to be minimized during joint optimization over training samples $(x_1, y_1), ...(x_n, y_n)$ is as follows:

$$\min_{\theta} \sum_{i=1}^{n} l_c(x_i, y_i; \theta_m, \theta_b) + l_s(x_i; \theta_s, \theta_b) \tag{3}$$

This is a multi-task learning pipeline, the losses of the two tasks are added together. The gradients of the backbone network are updated according to both of them.

In the stage of Test-Time Training, a single test sample $x$ is used to minimize the self-supervised loss. The parameters of the backbone network and the self-supervised head are updated as follows.

$$\min_{\theta} l_s(x; \theta_s, \theta_b) \tag{4}$$

Then, classification prediction $\hat{y}$ is obtained from the input $x$ with the updated network parameters $\theta^* = (\theta_b^*, \theta_m)$. Test-time training method claims that the minimization over $\theta_b$ or both $\theta_b, \theta_s$ is almost the same [34]. The difference only exists when doing more than one gradient optimization.

*2) Test-time training for visual odometry:* In this paper, test-time training for visual odometry is different from the image recognition problem. The proposed method has two stages.

The first stage is the depth pre-training. The pre-trained depth network is formulated as $\theta = (\theta_1, ..., \theta_n)$. In the pre-training stage, the main task is the supervised disparity (depth) prediction. In this work, there is a self-supervised stereo matching loss using the same disparity prediction $\hat{d}_i$. Assuming the image and disparity pairs $x_i, \overline{d}_i$, the optimization formulation for this stage is as follows.

$$\min_{\theta} \sum_{i=1}^{n} l_m(x_i, \hat{d}_i, \overline{d}_i; \theta) + l_s(x_i, \hat{d}_i; \theta) \tag{5}$$

In test-time training for visual odometry, the main task is the supervised depth estimation with the sparse ground truth annotations. The self-supervised auxiliary task is self-supervised stereo matching.

In detail, the loss functions of the main task and self-supervised auxiliary task are shown as follows.

$$\begin{aligned} l_m(x_i, \hat{d}_i, \overline{d}_i; \theta) &= |\hat{d}_i - \overline{d}_i| \\ l_s(x_i, \hat{d}_i; \theta) &= |W(x_i^R, \hat{d}_i^L) - x_i^L| \end{aligned} \tag{6}$$

Where $x_i^L, x_i^R$ refer to the left and the right of the calibrated stereo images. $\hat{d}_i^L$ is the predicted disparity map of the left view. $W()$ is a stereo image-warping operation.

The second stage is test-time training on the visual odometry videos $v = (x_1, ..., x_n)$. The pre-trained parameters $\theta$ of the depth network are updated as the following formulation.

$$\min_{\theta} \; l_s(v_i, \hat{d}_i; \theta) \tag{7}$$

For new each video frame $v_i$, the updated network parameters are denoted as $\theta^*$. The predicted disparity $\hat{d}_i$ is obtained with parameters $\theta^*$. Same as the standard TTT, one gradient step is performed. According to the previous works [34], the single-iteration update does not suffer the difference between the minimization over $\theta_b, \theta_s$ and the minimization over $\theta_b$.

*3) Sequential test-time training for visual odometry:* Given the nature of the visual odometry task, a Sequential Test-Time Training (SeqTTT) strategy is proposed to produce better depth prior. For a video sequence, frames within a local video clip share similar information and belong to close data domains. Drawing inspiration from this, the optimization of frame $t$ in each short video clip $c_i$ is initialized using the parameters $\theta_{t-1}$ of the previous frame $t-1$, rather than the pre-trained depth network parameters $\theta$. Only a short video clip satisfies the theoretical guarantee of test-time training. Longer

video clips can result in significant differences between the optimization goals in Eq. 7 and in Eq. 5.

## III. Experiments

### A. Details

For the visual odometry experiments, the depth network is pre-trained on two still-image datasets: KITTI Depth [1] and SceneFlow [2], and evaluated on KITTI odometry [3] and VKITTI2 [4]. The image of KITTI Depth is cropped into $256 \times 800$. The image of SceneFlow is cropped to a resolution of $320 \times 512$. The cropping is aligned with the bottom for the vertical dimension and with the center for the horizontal dimension. For the depth map, only pixels with depth values in the range of $[1, 100]$ are considered. In the ablation studies of test-time training, ResNet18 [32] and MobileNetv2 [33] are used on the KITTI Depth and SceneFlow datasets separately. AdamW optimizer with a learning rate of 5e-6 is used to perform Test-Time Training, while the same optimizer with a learning rate of 1e-3 is used in the pre-training stage. Experiments are conducted using Nvidia A40 GPU.

**Depth metrics:** rel sqr: relative square root error on each pixel. rmse log: logarithm root mean squared error. Depth accuracy: The percent of correct predictions. The pixel errors lower than $\{1.25, 1.25^2, 1.25^3\}$ are defined as correct predictions. **stereo disparity metrics:** EPE(all): every pixel error on all pixels. EPE(occ): every pixel error on stereo occlusion area. MACs: Multiply–Accumulate Operations. **Odometry metrics:** $t_{err}, r_{err}$: KITTI sequential translation and rotation error. RPE: relative pose error. ATE: total absolute trajectory error.

### B. Stereo network with adaptive disparity

In this part, state-of-the-art results on real-world benchmarks are shown to demonstrate the advantage of the new depth network. As shown in Tab. I, this network can achieve state-of-the-art results on KITTI depth Eigen split benchmark. Firstly, these results suggest that the stereo-based depth estimation has a significant advantage compared with the monocular depth estimation networks. Secondly, the proposed network shows higher accuracy compared with the state-of-the-art stereo depth estimation methods. Fig. 3 shows the advantages of this work over the previous, especially on the details of roadside objects.

Then, there are more ablation studies on SceneFlow dataset [31]. This dataset is a larger-scale simulation dataset, which provides dense ground truth disparity labels (Fig. 4) and stereo occlusion labels to measure the predicted disparity maps. Meanwhile, the simulation dataset covers more complex scenarios. The dense disparity quality can be directly measured with every pixel error (EPE).

[1] www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction
[2] lmb.informatik.uni-freiburg.de/resources/datasets/SceneFlowDatasets.en.html
[3] www.cvlibs.net/datasets/kitti/eval_odometry.php
[4] europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds-vkitti-2/

| Method | Depth Error | | Depth Accuracy (%) | | |
|---|---|---|---|---|---|
| | rel sqr | rmse log | $< 1.25$ | $< 1.25^2$ | $< 1.25^3$ |
| Monocular network | | | | | |
| Adabins[9] | 0.1900 | 0.0880 | 96.4 | 99.5 | 99.9 |
| iDisc [10] | 0.1450 | 0.0770 | 97.7 | 99.7 | 99.9 |
| URCDC-Depth [11] | 0.1420 | 0.0760 | 97.7 | 99.7 | 99.9 |
| SwinV2MIM [12] | 0.1390 | 0.0750 | 97.7 | 99.8 | 100.0 |
| Stereo Network | | | | | |
| PSMNet [6] | 0.0447 | 0.040 | 99.6 | 99.9 | 100.0 |
| CascadePSM [15] | 0.0542 | 0.042 | 99.7 | 99.9 | 100.0 |
| CascadeGWC [15] | 0.0695 | 0.048 | 99.5 | 99.9 | 99.9 |
| IGEVStereo [7] | 0.0600 | 0.041 | 99.6 | 99.9 | 99.9 |
| This work | **0.0405** | **0.036** | **99.8** | **100.0** | **100.0** |

TABLE I
Depth estimation results evaluated on KITTI depth (Eigen split) still image dataset.

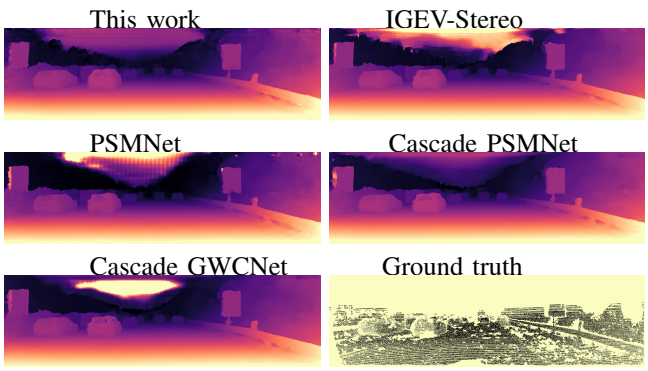



Fig. 3. Comparison of state-of-the-art stereo depth estimation results.

| Model | Encoder | EPE(all) | EPE(occlusion) | MACs | Params |
|---|---|---|---|---|---|
| PSMNet | PSMNet Encoder [6] | 2.58 | 7.38 | 776.33G | 4.06M |
| PSMNet | ResNet18 [32] | **2.26** | **6.96** | 966.55G | 1.57M |
| PSMNet | MobileNetv2 [32] | 2.36 | 7.16 | 517.97G | **0.74**M |

TABLE II
Experiment results of using pre-trained general light-weighted network. $*$ denotes that PSMNet does not use the default cascaded head.

| Searching Space | Encoder | EPE(all) | EPE(occlusion) |
|---|---|---|---|
| Pre-defined | MobileNetv2 | 2.36 | 7.16 |
| AdaSearch | MobileNetv2 | **1.97** | **4.81** |
| Pre-defined | ResNet18 | 2.26 | 6.96 |
| AdaSearch | ResNet18 | **1.86** | **4.63** |
| Pre-defined | PSMNet Encoder | 2.58 | 7.38 |
| AdaSearch | PSMNet Encoder | **1.99** | **4.68** |

TABLE III
Experiment results of adaptive disparity searching space.

*1) Advanced encoder :* In the proposed depth network, the impact of the feature encoder network is analyzed. As shown in Tab. II, the results of PSMNet [6] are compared by using PSMNet Encoder and the other foundational networks, e.g. ResNet18 [32], MobileNetv2 [33], which are more efficient. The results are positive using the general encoder networks [32], [33] to replace the original feature encoder in the two-stage stereo network [6].

*2) Adaptive disparity searching space:* We present an experiment that compares the proposed adaptive disparity searching space (AdaSearch) method with the previous pre-defined disparity searching space method [6], [14]. AdaSearch

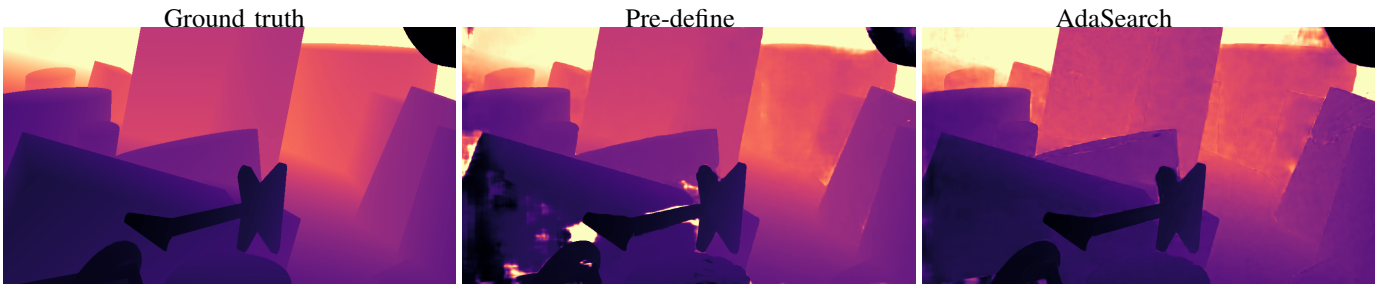



Fig. 4. Visualization of the improvement with adaptive adaptive disparity searching space using ResNet encoder.

| Test set | Training set | T-T-T | Depth Error Metrics | | | | Depth Accuracy Metric(%) | | | Camera Pose Error Metric(lower) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | abs rel | rel sqr | rmse | rmse log | $\tau < 1.25$ | $< 1.25^2$ | $< 1.25^3$ | $t_{err}$(%) | $r_{err}$ (deg/100m) | $RPE_{tran}$ (m) | $RPE_{rot}$ (deg) |
| KITTIOdometry09 | (Video)K-Odom 0-8 | ✗ | 0.1519 | 4.4338 | 6.6678 | 0.2760 | 89.34 | 93.64 | 95.87 | 2.53 | 1.06 | 0.034 | 0.048 |
| | (Image)SenceFlow | ✓ | 0.0877 | 1.1358 | 3.9558 | 0.1679 | 93.21 | 97.04 | 98.38 | 3.56 | 2.40 | 0.044 | 0.059 |
| | (Image)KITTIDepth | ✓ | **0.0669** | **0.4872** | **3.3448** | **0.1515** | **93.83** | **97.09** | **98.49** | **2.01** | **1.35** | **0.018** | **0.041** |
| KITTIOdometry10 | (Video)K-Odom 0-8 | ✗ | 0.2079 | 5.7366 | 6.7805 | 0.3607 | 85.17 | 90.23 | 93.29 | 2.65 | 1.46 | 0.024 | 0.050 |
| | (Image)SenceFlow | ✓ | 0.1550 | 3.8829 | 4.4014 | 0.2294 | 90.60 | 95.25 | 97.15 | 2.31 | 1.52 | **0.028** | **0.048** |
| | (Image)KITTIDepth | ✓ | **0.0768** | **0.5115** | **2.8574** | **0.1692** | **92.41** | **96.25** | **98.08** | **2.01** | **1.32** | 0.023 | 0.045 |
| VKTTI2-20 | (Video)VKTTI2- 1,2,6,18 | ✗ | 0.7648 | 84.2296 | 62.1419 | 0.6397 | 78.75 | 84.38 | 87.56 | 13.90 | 3.75 | 0.066 | 0.061 |
| | (Image)SceneFLow | ✓ | **0.1132** | **18.9506** | **37.9738** | **0.2465** | **91.14** | **94.72** | **96.56** | **8.62** | **2.16** | **0.061** | **0.046** |
| | (Image)KITTIDepth | ✓ | 0.1227 | 20.6927 | 43.7050 | 0.2590 | 90.35 | 94.14 | 96.16 | 9.63 | 2.94 | 0.063 | 0.054 |

TABLE IV
RESULTS OF TRAIN-FROM-SCRATCH AND TEST-TIME TRAINING ON THE KITTI VISUAL ODOMETRY DATASET.

is realized with a monocular network. The results presented in Tab. III demonstrate that the AdaSearch method significantly outperforms the default method. These findings suggest that learning disparity knowledge from monocular features is useful. In addition, errors of stereo occlusion regions become much lower, which suggests that this method can improve the stereo disparity of difficult occlusion regions. Fig. 4 also shows the improvement using adaptive disparity searching space. It can solve the stereo occlusion problem well.

### C. Test-time training V.S. Training on the target dataset

In this experiment, hybrid dense direct visual odometry [3] is used. Baseline models are trained on target datasets, KITTI odometry or VKITTI2. Test-time training models are pre-trained on KITTI depth image dataset or SceneFlow simulation dataset. Then, they are tested on the same test dataset. Finally, the test-time training method outperforms the baseline in terms of both odometry and depth results, as indicated in Table IV. This suggested that the test-time training method can achieve a good performance and save training costs.

Moreover, the results of simulation dataset (SceneFlow) are also better than the baseline approach. This suggests that the proposed method is robust and generalized even with a large data domain.

### D. Ablation studies

Additionally, to show the impact of test-time training method, the results of three cases (test with pre-training, image-level test-time training, and sequential test-time training) are compared on the KITTI odometry dataset. Firstly, the efficiency of the proposed method is analyzed. Secondly, there are experiments to explore the domain gap between image data and video data. Thirdly, the domain gap between the simulation data and real-world data is explored. Then, a more detailed experiment is performed to find the suitable number of video frames for updating sequential test-time training.

| | Test-time training | Train from scratch |
|---|---|---|
| Online-Optim | 56.9ms/f x 1591frames | ✗ |
| Offline-Training | ✗ | 0.1s/i x 40000 iters |
| Total training cost | 91s | 4020s |
| Depth/frame | 18.8ms | 15.5ms |
| Pose/frame | 22.8ms | 26.3ms |
| Total inference/frame | 108.2ms | 52.1ms |

TABLE V
INFERENCE TIME AND TRAINING TIME COST ON KITTI ODOMETRY SEQ09 ON NVIDIA A40 GPU. THE TIME IS MEASURED USING A TWO-STAGE DEPTH ESTIMATION NETWORK WITH MOBILENETV2 ENCODER [28].

*1) Inference time and training cost:* The inference time and training cost are analyzed here. Tab. V suggests that online optimization with test-time training method increase the inference time. However, with this strategy, the total training cost on the whole video sequence can be significantly decreased. Meanwhile, the competitive accuracy and error results on depth and camera pose can be found in Tab. IV.

*2) Domain gap between image data and video data:* There are experiments (Tab. VI) to evaluate test-time training on the KITTI odometry 11 video sequences.

These results demonstrate that utilizing test-time training (TTT) leads to better depth prior and better pose estimation results than the baseline. Furthermore, sequential TTT outperforms the standard image-level TTT. However, it is noteworthy that some depth metrics do not consistently show better results than image-level TTT, this maybe because the depth is evaluated with sparse Lidar depth annotations.

*3) Domain gap between the simulation data and real-world data:* Here, a popular simulation dataset, SceneFlow, is used as the pre-training dataset. the results on KITTI odometry 11 video sequences are depicted in Tab. VII. The results suggest that the sequential TTT strategy also performs best

| SeqID | TTT | Depth Error Metrics(lower) | | | | Depth Accuracy Metric(%)(higher) | | | Camera Pose Error Metric(lower) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | abs rel | rel sqr | rmse | rmse log | $\tau < 1.25$ | $< 1.25^2$ | $< 1.25^3$ | $t_{err}(\%)$ | $r_{err}$ (deg/100m) | $RPE_{tran}$ (m) | $RPE_{rot}$ (deg) |
| 00 | ✗ | 0.0753 | 0.4788 | 3.2412 | 0.1730 | 92.35 | 96.16 | 97.88 | 3.56 | 1.52 | 0.035 | 0.065 |
| | TTT | 0.0752 | 0.4773 | 3.2346 | 0.1728 | 92.37 | 96.16 | 97.88 | 3.52 | 1.50 | 0.035 | 0.064 |
| | SeqTTT | **0.0752** | 0.4826 | **3.2153** | **0.1723** | **92.41** | **96.18** | **97.89** | **3.32** | **1.39** | **0.035** | **0.064** |
| 01 | ✗ | 0.0912 | 1.5150 | 6.1106 | 0.2118 | 92.19 | 95.82 | 97.60 | 10.12 | 1.83 | 0.217 | 0.123 |
| | TTT | 0.0911 | 1.5124 | 6.1000 | 0.2117 | 92.23 | 95.83 | 97.60 | 10.09 | 1.81 | 0.217 | 0.122 |
| | SeqTTT | **0.0901** | **1.4851** | **6.0236** | **0.2109** | **92.41** | **95.83** | 97.59 | **9.96** | **1.78** | **0.217** | **0.121** |
| 02 | ✗ | 0.0580 | 0.3114 | 2.6845 | 0.1184 | 95.81 | 98.23 | 99.16 | 4.05 | 2.20 | 0.063 | 0.062 |
| | TTT | 0.0578 | 0.3088 | 2.6755 | 0.1182 | 95.83 | 98.24 | 99.16 | 3.97 | 2.15 | 0.063 | 0.062 |
| | SeqTTT | **0.0572** | **0.2982** | **2.6403** | **0.1177** | **95.88** | **98.24** | **99.16** | **3.57** | **1.93** | **0.062** | **0.060** |
| 03 | ✗ | No label available | | | | | | | 4.51 | 3.70 | 0.037 | 0.050 |
| | TTT | | | | | | | | 4.42 | 3.64 | 0.037 | 0.050 |
| | SeqTTT | | | | | | | | **3.96** | **3.32** | **0.035** | **0.048** |
| 04 | ✗ | 0.0693 | 0.5052 | 3.6008 | 0.1342 | 94.89 | 98.00 | 98.92 | 3.50 | 3.95 | 0.049 | 0.069 |
| | TTT | 0.0689 | 0.5000 | 3.5820 | 0.1338 | 94.94 | 98.00 | 98.92 | 3.37 | 3.88 | 0.048 | 0.068 |
| | SeqTTT | **0.0669** | **0.4767** | **3.4986** | **0.1319** | **95.14** | **98.02** | **98.92** | **2.64** | **3.54** | **0.044** | **0.063** |
| 05 | ✗ | 0.0834 | 0.6121 | 3.5515 | 0.1873 | 90.87 | 95.15 | 97.39 | 4.46 | 1.83 | 0.030 | 0.049 |
| | TTT | 0.0833 | **0.6118** | 3.5463 | 0.1872 | 90.89 | 95.15 | 97.39 | 4.41 | 1.81 | 0.030 | 0.049 |
| | SeqTTT | **0.0833** | 0.6284 | **3.5379** | **0.1870** | **90.95** | **95.16** | **97.39** | **4.20** | **1.72** | **0.029** | **0.048** |
| 06 | ✗ | 0.1009 | 1.5527 | 5.8067 | 0.2198 | 90.35 | 95.18 | 97.29 | 4.26 | 2.20 | 0.035 | 0.041 |
| | TTT | 0.1005 | 1.5412 | 5.7804 | 0.2195 | 90.42 | **95.18** | **97.29** | 4.17 | 2.16 | 0.035 | 0.041 |
| | SeqTTT | **0.0988** | **1.4892** | **5.6525** | **0.2186** | **90.72** | 95.16 | 97.28 | **3.64** | **1.88** | **0.034** | **0.039** |
| 07 | ✗ | **0.0746** | **0.4762** | 3.1128 | 0.1704 | 92.35 | 95.96 | 97.81 | 2.86 | 1.57 | 0.024 | 0.046 |
| | TTT | 0.0747 | 0.4810 | **3.1108** | **0.1704** | 92.36 | 95.96 | 97.81 | 2.82 | 1.56 | 0.024 | 0.046 |
| | SeqTTT | 0.0760 | 0.5245 | 3.1198 | 0.1708 | **92.36** | **95.96** | **97.81** | **2.53** | **1.46** | **0.023** | **0.045** |
| 08 | ✗ | 0.0848 | 0.7652 | 4.0976 | 0.1988 | 91.22 | 95.25 | 97.18 | 3.25 | 1.54 | 0.034 | 0.045 |
| | TTT | **0.0846** | 0.7602 | 4.0841 | 0.1985 | 91.25 | 95.26 | 97.19 | 3.21 | 1.51 | 0.034 | 0.045 |
| | SeqTTT | 0.0847 | **0.7527** | **4.0368** | **0.1980** | **91.33** | **95.27** | **97.19** | **2.99** | **1.40** | **0.034** | **0.044** |
| 09 | ✗ | 0.0675 | 0.5040 | 3.3834 | 0.1520 | 93.79 | 97.08 | 98.48 | 2.47 | 1.82 | 0.034 | 0.048 |
| | TTT | 0.0673 | 0.5004 | 3.3751 | 0.1518 | 93.80 | 97.09 | 98.48 | 2.44 | 1.79 | 0.034 | 0.048 |
| | SeqTTT | **0.0669** | **0.4872** | **3.3448** | **0.1515** | **93.83** | **97.09** | **98.49** | **2.01** | **1.35** | **0.018** | **0.041** |
| 10 | ✗ | 0.0771 | 0.4422 | 2.8378 | 0.1682 | 92.33 | 96.26 | 98.10 | 2.27 | 2.00 | 0.023 | 0.048 |
| | TTT | 0.0769 | **0.4414** | **2.8302** | **0.1680** | 92.35 | **96.27** | **98.11** | 2.24 | 1.97 | 0.023 | 0.048 |
| | SeqTTT | **0.0768** | 0.5115 | 2.8574 | 0.1692 | **92.41** | 96.25 | 98.08 | **2.01** | **1.32** | **0.023** | **0.045** |

TABLE VI

TEST-TIME TRAINING RESULTS WITH THE DEPTH NETWORK PRE-TRAINED ON THE KITTI DEPTH DATASET, AND EVALUATED ON THE KITTI ODOMETRY VIDEOS. THE DEPTH ANNOTATIONS OF THE LIDAR DATA OF SEQUENCE 03 ARE MISSED.

| seqID | TTT | Depth Error Metrics | | | | Depth Accuracy Metric | | | Camera Pose Error Metric | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | abs rel | rel sqr | rmse | rmse log | $\tau < 1.25$ | $< 1.25^2$ | $< 1.25^3$ | $t_{err}(\%)$ | $r_{err}$ (deg/100m) | $RPE_{tran}$ (m) | $RPE_{rot}$ (deg) |
| 00 | ✗ | 0.1658 | 4.9154 | 5.2438 | 0.2397 | 90.55 | 95.27 | 97.03 | 4.01 | 1.59 | 0.041 | 0.067 |
| | TTT | 0.1442 | 3.7824 | 4.7039 | 0.2211 | 91.23 | 95.70 | 97.36 | 3.94 | 1.57 | 0.040 | 0.067 |
| | SeqTTT | **0.1185** | **2.1272** | **4.2252** | **0.2164** | 91.20 | 95.54 | 97.23 | **3.66** | **1.55** | **0.039** | 0.068 |
| 01 | ✗ | 0.1569 | 3.6756 | 8.0197 | 0.2750 | 87.84 | 93.90 | 96.35 | 10.94 | 2.31 | 0.241 | 0.144 |
| | TTT | 0.1433 | 3.1990 | 7.7684 | 0.2608 | 88.76 | 94.35 | 96.64 | 10.42 | 2.20 | 0.232 | 0.139 |
| | SeqTTT | 0.1224 | 2.5743 | 7.3465 | 0.2404 | 90.47 | 95.01 | 96.96 | 10.20 | 2.08 | 0.227 | 0.132 |
| 02 | ✗ | 0.0920 | 1.3181 | 3.8623 | 0.1566 | 93.97 | 97.49 | 98.63 | 3.82 | 1.94 | 0.060 | 0.061 |
| | TTT | 0.0809 | 0.9311 | 3.5089 | 0.1413 | 94.63 | 97.88 | 98.91 | 3.70 | 1.87 | 0.059 | 0.060 |
| | SeqTTT | **0.0750** | **0.6905** | **3.2939** | **0.1329** | **95.04** | **98.06** | **99.01** | **3.40** | **1.73** | **0.058** | **0.060** |
| 03 | ✗ | No label available | | | | | | | 3.58 | 2.10 | 0.036 | 0.048 |
| | TTT | | | | | | | | 3.63 | 2.03 | 0.036 | 0.047 |
| | SeqTTT | | | | | | | | 3.30 | 1.78 | 0.034 | 0.047 |
| 04 | ✗ | 0.1095 | 1.4876 | 4.8488 | 0.1761 | 91.34 | 96.51 | 98.01 | 1.71 | 3.36 | 0.046 | 0.065 |
| | TTT | 0.0982 | 1.1318 | 4.5293 | 0.1639 | 92.08 | 96.98 | 98.38 | 1.83 | 3.29 | 0.045 | 0.061 |
| | SeqTTT | **0.0959** | **1.0378** | **4.4151** | **0.1628** | **92.67** | 96.96 | 98.29 | **1.36** | **2.94** | **0.042** | **0.056** |
| 05 | ✗ | 0.1651 | 4.5391 | 5.4327 | 0.2477 | 89.53 | 94.56 | 96.69 | 2.93 | 1.06 | 0.033 | 0.045 |
| | TTT | 0.1459 | 3.5851 | 4.9031 | 0.2294 | 90.18 | 94.98 | 97.01 | 2.87 | 1.04 | 0.033 | 0.045 |
| | SeqTTT | 0.1185 | 1.9678 | 4.3023 | 0.2159 | 90.44 | 95.08 | 97.10 | 2.70 | 1.05 | 0.030 | 0.044 |
| 06 | ✗ | 0.2011 | 6.4907 | 7.5232 | 0.2796 | 87.71 | 94.08 | 96.45 | 2.48 | 0.83 | 0.044 | 0.036 |
| | TTT | 0.1777 | 5.2747 | 7.0128 | 0.2604 | 88.58 | 94.59 | 96.83 | 2.15 | 0.84 | 0.042 | 0.038 |
| | SeqTTT | **0.1496** | **3.7575** | **6.4927** | **0.2453** | **89.56** | **94.97** | **96.96** | **1.95** | **0.99** | **0.039** | 0.042 |
| 07 | ✗ | 0.2100 | 7.2782 | 5.5129 | 0.2589 | 89.83 | 94.66 | 96.69 | 3.12 | 1.37 | 0.029 | 0.045 |
| | TTT | 0.1864 | 6.1665 | 4.9813 | 0.2396 | 90.60 | 95.12 | 97.04 | 3.02 | 1.40 | 0.028 | 0.045 |
| | SeqTTT | 0.1532 | 4.1315 | 4.4043 | 0.2252 | 91.03 | 95.31 | 97.18 | 2.75 | 1.36 | 0.025 | 0.044 |
| 08 | ✗ | 0.2156 | 6.9627 | 6.6160 | 0.2932 | 87.86 | 93.33 | 95.60 | 3.01 | 1.14 | 0.038 | 0.044 |
| | TTT | 0.1875 | 5.6046 | 6.0142 | 0.2701 | 88.83 | 94.01 | 96.14 | 2.91 | 1.11 | 0.038 | 0.044 |
| | SeqTTT | **0.1505** | **3.4666** | **5.2479** | **0.2429** | **89.68** | **94.57** | **96.57** | **2.62** | **1.02** | **0.036** | **0.043** |
| 09 | ✗ | 0.1154 | 2.3646 | 4.7140 | 0.1976 | 91.96 | 96.36 | 97.90 | 3.82 | 2.41 | 0.046 | 0.060 |
| | TTT | 0.0994 | 1.6901 | 4.2581 | 0.1789 | 92.73 | 96.82 | 98.23 | 3.67 | 2.36 | 0.045 | 0.059 |
| | SeqTTT | 0.0878 | 1.1359 | 3.9569 | 0.1680 | 93.21 | 97.04 | 98.38 | 3.56 | 2.40 | 0.044 | 0.059 |
| 10 | ✗ | 0.2542 | 9.6267 | 6.0670 | 0.2936 | 88.90 | 93.93 | 96.08 | 2.43 | 1.61 | 0.032 | 0.050 |
| | TTT | 0.2185 | 7.8458 | 5.4104 | 0.2674 | 89.91 | 94.67 | 96.66 | 2.38 | 1.51 | 0.030 | 0.049 |
| | SeqTTT | **0.1550** | **3.8829** | **4.4014** | **0.2294** | **90.60** | **95.25** | **97.15** | **2.31** | 1.52 | **0.028** | **0.048** |

TABLE VII

TEST-TIME TRAINING RESULTS WITH THE DEPTH NETWORK PRE-TRAINED ON SIMULATION DATASET, SCENEFLOW.

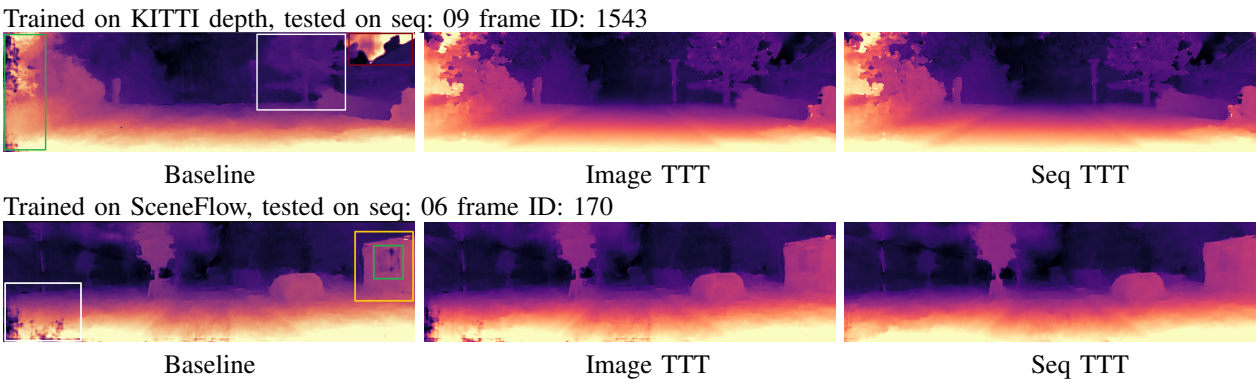
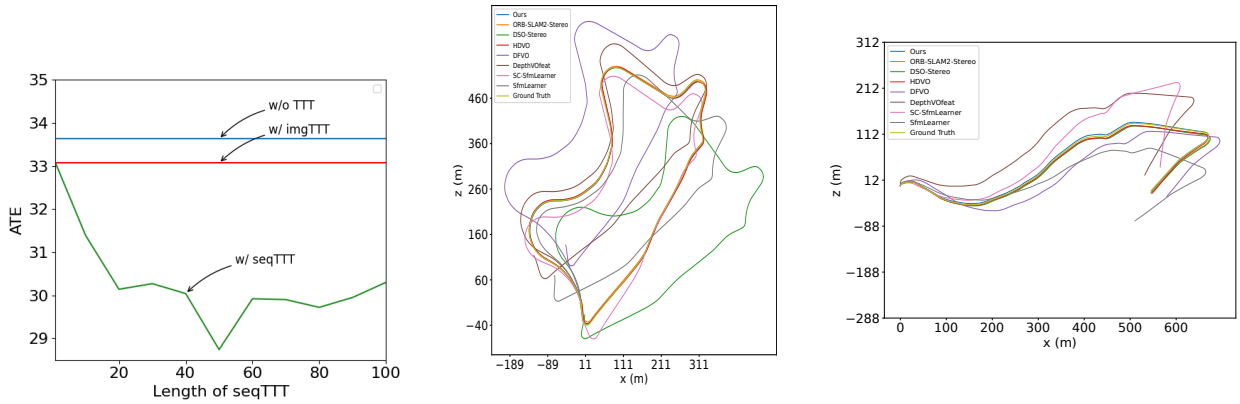



Fig. 5. The visualization corresponding to experiments in Tab. VI and Tab. VII.




a. b. c.

Fig. 6. a: The ablation for the length of the sequential test-time training. b,c: Comparison with other methods on KITTI sequence 09 and 10

with the simulation data pre-training. And these results are also competitive compared with the results using the pre-training of real-world data in Tab. VI. The visual improvement of the depth map can be identified in Fig. 5.

*4) Sequential test-time training for every N frames:* To support the theory analysis of the sequential Test-Time Training, i.e. the video clip should keep a short length, there are experiments to compare the visual odometry results (accumulated Absolute Trajectory Error) with different video clip lengths in sequential Test-Time Training. As Fig. 6 suggests, the lowest visual odometry ATE is obtained when *seqTTT* re-initializes the model parameter every 50 frames in KITTI odometry dataset. Re-initializing network parameters with a longer clip ($>50$ frames) is worse as shown in green line.

### *E. Comparison with state-of-the-art visual odometry methods*

To show the advanced performance of the proposed method, it is compared with previous state-of-the-art methods, including traditional model-based, full deep learning, and hybrid methods, as shown in Table VIII. The proposed visual odometry method achieves competitive results compared to the previous. Notably, the proposed method is not trained on the KITTI odometry dataset, while the previous deep hybrid methods [3], [2], [1] are trained on the sequence 00-08 of this dataset, and their results are re-scaled with ground truth labels.

| Method | seq.9 | | seq.10 | |
|---|---|---|---|---|
| | $t_{err}$ | $r_{err}$ | $t_{err}$ | $r_{err}$ |
| **model-based methods** | | | | |
| ORBSLAM2-stereo [35] | 0.85 | 0.26 | 0.56 | 0.24 |
| DSO-stereo [36] | 41.04 | 14.47 | 1.34 | 0.42 |
| VISO2 [37] | 18.06 | 1.25 | 26.10 | 3.26 |
| **Deep learning methods** | | | | |
| SfmLearner [27] | 11.32 | 4.07 | 15.25 | 4.06 |
| DepthVOfeat [38] | 11.89 | 3.60 | 12.82 | 3.41 |
| SC-SfmLearner [39] | 7.64 | 2.19 | 10.74 | 4.58 |
| F2FPE [40] | 2.36 | 1.06 | 3.00 | 1.28 |
| **hybrid methods** | | | | |
| UnOS [1] | 5.21 | 1.80 | 5.20 | 2.18 |
| DFVO [2] | 2.07 | 0.23 | 2.06 | 0.36 |
| HDVO [3] | 1.97 | 0.71 | 1.89 | 0.56 |
| This work | **0.81** | 0.44 | **1.54** | 0.73 |

TABLE VIII
COMPARISON WITH THE PREVIOUS METHODS USING KITTI METRICS $t_{err}, r_{err}$ ON SEQ 09, 10 OF KITTI ODOMETRY VIDEO SEQUENCES. WE ONLY USE HUBER LOSS FOR IMPROVING ODOMETRY RESULTS, WITHOUT OTHER AUGMENTATIONS, E.G. LOOP CLOSURE, AND OTHER GLOBAL OPTIMIZATION.

Additionally, the results on KITTI video sequences 09 and 10 are visualized and compared in Fig. 6 b,c. The proposed method achieves the best trajectory predictions without extra training on the target dataset and loop closure etc.

## IV. Conclusion

This paper explores the domain gap problem in hybrid visual odometry methods. The disparity gap of different images is solved by a new stereo network with adaptive disparity. The domain gap across different datasets is solved by the sequential test-time training on hybrid visual odometry. There are significant improvements in the accuracy of depth estimation and visual odometry. And state-of-the-art results can be achieved without re-training on the new dataset. Future investigations will focus on improving the efficiency of this test-time training method in hybrid SLAM system.

## References


[1] Y. Wang, P. Wang, Z. Yang, C. Luo, Y. Yang, and W. Xu, "Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos," in *CVPR*. IEEE, 2019, pp. 8071–8081.

[2] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?" in *ICRA*. IEEE, 2020, pp. 4203–4210.

[3] Z. Liu, E. Malis, and P. Martinet, "A new dense hybrid stereo visual odometry approach," in *IROS*. IEEE, 2022, pp. 6998–7003.

[4] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *CVPR*. IEEE, 2020, pp. 1281–1292.

[5] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *ECCV*. Springer, 2018, pp. 817–833.

[6] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *CVPR*. IEEE, 2018, pp. 5410–5418.

[7] G. Xu, X. Wang, X. Ding, and X. Yang, "Iterative geometry encoding volume for stereo matching," in *CVPR*, 2023.

[8] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*. IEEE, 2019, pp. 3828–3838.

[9] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *CVPR*, 2021, pp. 4009–4018.

[10] L. Piccinelli, C. Sakaridis, and F. Yu, "idisc: Internal discretization for monocular depth estimation," in *VPR*, 2023, pp. 21 477–21 487.

[11] S. Shao, Z. Pei, W. Chen, R. Li, Z. Liu, and Z. Li, "Urcdc-depth: Uncertainty rectified cross-distillation with cutflip for monocular depth estimatione," *IEEE Transactions on Multimedia*, 2023.

[12] Z. Xie, Z. Geng, J. Hu, Z. Zhang, H. Hu, and Y. Cao, "Revealing the dark secrets of masked image modeling," in *CVPR*, 2023, pp. 14 475–14 485.

[13] 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 *CVPR*. IEEE, 2016, pp. 4040–4048.

[14] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *ICCV*. IEEE, 2017, pp. 66–75.

[15] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2495–2504.

[16] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *CVPR*, 2019, pp. 3273–3282.

[17] L. Lipson, Z. Teed, and J. Deng, "Raft-stereo: Multilevel recurrent field transforms for stereo matching," in *International Conference on 3D Vision*, 2021.

[18] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, and S. Liu, "Practical stereo matching via cascaded recurrent network with adaptive correlation," in *CVPR*, 2022, pp. 16 263–16 272.

[19] Z. Liu, E. Malis, and P. Martinet, "One-stage deep stereo network," in *ICASSP*, 2024.

[20] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *ICML*. PMLR, 2020, pp. 9229–9248.

[21] Y. Liu, P. Kothari, B. Van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, "Ttt++: When does self-supervised test-time training fail or thrive?" *NeuIPS*, vol. 34, pp. 21 808–21 820, 2021.

[22] Y. Gandelsman, Y. Sun, X. Chen, and A. Efros, "Test-time training with masked autoencoders," *NeuIPS*, vol. 35, pp. 29 374–29 385, 2022.

[23] M. Zhang, S. Levine, and C. Finn, "Memo: Test time robustness via adaptation and augmentation," *NeuIPS*, vol. 35, pp. 38 629–38 642, 2022.

[24] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6028–6039.

[25] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *ICLR*, 2021.

[26] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*. IEEE, 2017, pp. 270–279.

[27] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*. IEEE, 2017, pp. 1851–1858.

[28] A. Bangunharcana, J. W. Cho, S. Lee, I. S. Kweon, K.-S. Kim, and S. Kim, "Correlate-and-excite: Real-time stereo matching via guided cost volume excitation," in *Proceedings of International Conference on Intelligent Robots and Systems*. Prague, Czech Republic: IEEE, 2021, pp. 3542–3548.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.

[30] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

[31] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*. IEEE, 2015, pp. 2758–2766.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[34] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *ICML*. PMLR, 2020, pp. 9229–9248.

[35] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *TRO*, vol. 33, no. 5, pp. 1255–1262, 2017.

[36] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *IROS*. IEEE, 2015, pp. 1935–1942.

[37] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *IV*. IEEE, 2011, pp. 963–968.

[38] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *CVPR*, 2018, pp. 340–349.

[39] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," *NeurIPS*, vol. 32, 2019.

[40] X. Li, Y. Hou, P. Wang, Z. Gao, M. Xu, and W. Li, "Transformer guided geometry model for flow-based unsupervised visual odometry," *Neural Computing and Applications*, pp. 1–12, 2021.