

# HDVO: Hybrid Dense Direct Visual Odometry

Ziming Liu<sup>1,2,3</sup>, Ezio Malis<sup>1</sup>, *Member, IEEE*, Philippe Martinet<sup>1</sup>, *Member, IEEE*

**Abstract**—Dense direct visual odometry (DDVO) is a well-known robust model-based visual odometry method, that computes robot self-localization with visual data. However, to obtain a metric localization depth information is needed, which is not available in most applications, especially in outdoor scenarios. For this reason, we focus on a hybrid dense direct visual odometry (HDVO) that combines a model-based localization module and a data-based stereo depth estimation module. Since depth is estimated from a stereo sensor, the proposed HDVO can achieve better localization performance than methods using only monocular RGB data. Moreover, the learning part of HDVO is optimized with the photometric loss of DDVO without any depth supervision signal. For this loss, occlusion and low-texture will affect the optimization convergence. This paper investigates a binary mask based on stereo-temporal consistency to address the occlusion problem, along with another binary mask based on local patch consistency to solve the low-texture problem. Finally, it is shown that HDVO can achieve a significant advantage in terms of both inference speed and odometry error on public benchmarks. The corresponding code is available at <https://gitlab.inria.fr/acentauri-public/hdvo>.

**Index Terms**—Visual odometry, hybrid AI method, data-based algorithm, model-based algorithm, computer vision, robotics, autonomous driving.

## I. INTRODUCTION & RELATED WORKS

MODERN autonomous driving can use several sensors for car localization when the GPS signal is poor or absent: cameras, lidars, and sonars. Full vision-based methods have the advantage of having low cost and good performance. The main problem in visual odometry algorithms lies in accurately matching detected features or image intensities, which is a common problem for both feature-based visual odometry methods and direct visual odometry methods.

Feature-based VO method heavily relies on feature detection. Errors in feature detection and extraction cannot be corrected and can lead to subsequent feature matching failures [1], [2]. In contrast, intensity matching is included in the loop for pose estimation in direct visual odometry methods. Therefore, matching errors can be corrected in subsequent loop iterations, and direct methods exhibit more reliable and robust performance across various scenarios [3], [4].

Direct visual odometry methods can be categorized into dense and sparse methods. Dense direct visual odometry utilizes photometric intensities and may use geometric priors to estimate dense or semi-dense geometry [3], [5]. Additionally,

semi-dense direct visual odometry [6] introduces uncertainty weights to mitigate the effects of noisy pixels. These dense or semi-dense methods employ geometry priors, often referred to as depth prior, to optimize the model parameters in order to reduce computation time. Direct sparse visual odometry (DSO) [4] uses less data and does not need to rely on geometry priors since it can optimize all model parameters, including camera poses, camera intrinsic parameters, and geometry parameters (inverse depth), without incorporating geometry priors. On one hand, DSO decreases computation time by selectively using a set of independent regions around special points, such as corners. On the other hand, the computation time of optimization is increased since the depth is also unknown. Finally, the performance of direct visual odometry heavily relies on depth prior, pose prior, and the optimization method determining a trade-off between precision [3] and efficiency [4].

Considering the importance of priors in model-based visual odometry methods, hybrid visual odometry approaches are gaining increasing attention. These methods combine deep learning networks with model-based pose estimators. For instance, D3VO [7] and DVSO [8] employ learned depth and camera pose as geometry and pose priors for DSO module [4]. UnOS [9] proposes rigid-aware direct odometry (RDVO), which performs odometry estimation based on the estimated optical flow in rigid areas instead of relying on photometric loss. Another hybrid method, DF-VO [10] combines deep learning with epipolar geometry and Perspective-n-Point (PnP) method. Although these methods show impressive accuracy with deep learning. There are several critical problems. Firstly, they obtain depth priors using monocular solutions, which are not reliable in the depth scale. Secondly, it is difficult to achieve real-time inference with multiple deep learning networks.

All the previous hybrid visual odometry approaches involve a pre-trained depth estimation network, a camera pose estimation network or an optical flow network. The pre-trained depth estimation network is the most important part of these methods. Currently, only monocular networks are being explored. These monocular networks are trained with different loss functions, including weakly-supervised [11], semi-supervised [8], or unsupervised [12], [13] loss functions. Specifically, the photometric error is constructed by predicting the warped stereo image or the warped temporal image. In the case of stereo-warping, monodepth [14] introduces the stereo-matching constraint to optimize the depth. This constraint is widely utilized in subsequent work, often in combination with the temporal-warping method [12]. As for temporal-warping, SFMLearner [13] is one of the earliest networks that employed a temporal-warping-based loss. They proposed using a full

Ziming Liu was with <sup>1</sup> ACENTAURI team, Inria Center of Université Côte d'Azur, Sophia Antipolis, France. This project was funded by <sup>2</sup> 3IA Côte d'Azur, Université Côte d'Azur (Nice, France), and has been carried out at Inria. He is now with <sup>3</sup> Bosch Research, Shanghai, China. Email: [liuziming.email@gmail.com](mailto:liuziming.email@gmail.com)  
Ezio Malis and Philippe Martinet are with <sup>1</sup> ACENTAURI team, Inria Center of Université Côte d'Azur, Sophia Antipolis, France. Email: `{first name}. {name}@inria.fr`

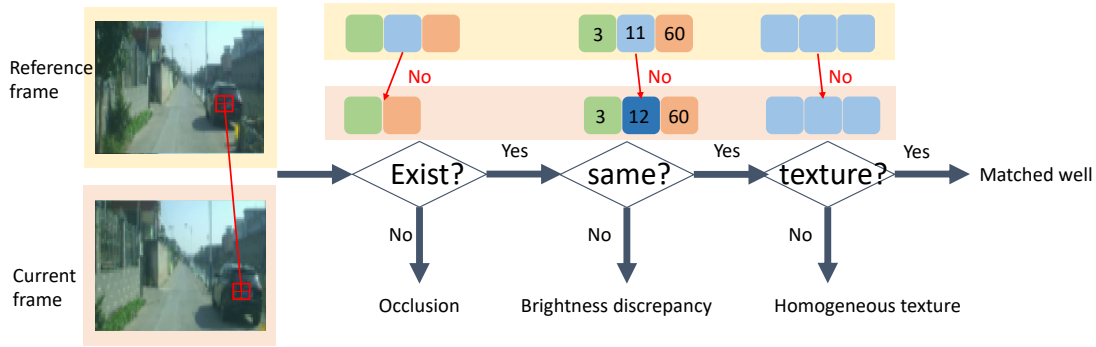


Fig. 1: Dense photometric loss matrix is affected by noisy pixels, including occlusion, brightness discrepancy, and homogeneous texture. Each colored cell represent one pixel in loss matrix. Different colors have different loss values. Three simple examples are shown above.

deep network to estimate camera poses, and then depth and camera pose are used for temporal warping. Similarly to dense direct visual odometry, temporal-warping methods also minimize photometric errors of the warped reference image. These temporal-warping methods predict camera pose with pose networks. Because the predictions of pose networks are not reliable, DVODepth [15] uses a direct visual odometry module to fine-tune the camera pose further.

However, these monocular networks have problems of poor generalization, which can be seen from two aspects. Firstly, monocular solutions can not recover the depth scale as the stereo vision in theory. This problem will be more serious in a different scene of the training data.

Secondly, monocular networks have poor generalization ability in new data domains, because they predict depth from a monocular image. In contrast, stereo networks are designed to learn stereo-matching relations. In previous works, stereo networks have shown better and more robust performance than monocular networks in both supervised [16] and weakly-supervised ways [11].

The objective of this paper is to address the issue of visual odometry, which is a crucial component in localization.

Considering the advantages of dense direct visual odometry and stereo depth estimation network, this paper proposes a new hybrid dense direct visual odometry (HDVO) approach. HDVO has an end-to-end structure. In the training stage, the data-based depth network is optimized with the same photometric loss of the dense direct visual odometry module.

However, this photometric loss of the DDVO faces many noisy pixels because of camera motion [7], [12], [17], [13]. The photometric loss matrix is actually the matching error between reference and current frames. Various masking methods have been explored to solve this problem. These masking methods can be divided into binary mask methods, which remove noisy pixels from the loss, and soft mask methods, which re-weight the influence of noisy pixels in the loss. Based on pixel matching, we can systemically summarize these problems into occlusion, brightness discrepancy, and homogeneous texture. Fig. 1 explains how these noisy pixels affect the photometric loss. Pixels in these ill-conditioned areas act as outliers for the photometric loss.

For binary masks, previous methods have relied on hand-crafted designs since the learnable methods are not reliable when ground truth masks are not available. These masks are designed based on different measurements, but they only address a portion of the aforementioned problems. For example, the principled mask [17] computes a binary mask based on image warping alignment, filtering out pixels that fall outside the image boundary. Monodepth2 [12] proposes an auto-mask based on whether adjacent frames have the same intensities. This mask filters out pixels from a static camera, relative static objects, or homogeneous texture areas. Additionally, an overlap mask and a blank mask are proposed [19], based on image warping alignment. DFNet and UnOS [20], [9] generate a binary mask using the forward-backward consistency of optical flow. Overall, these masks partially address some occlusion scenarios. And they only consider the information in the monocular sequence.

For soft masks, or namely "uncertainty" masks, previous works can be categorized into direct regularization methods and indirect methods. For instance, SfmLearner [13], an unsupervised depth and pose estimation framework, proposes an explainability mask that ranges from 0 to 1. This mask is optimized together with the loss by re-weighting the loss map, and it is regularized to avoid vanishing to 0. Similarly, "SFM from SFM" [21] introduces a probability uncertainty mask, which uses the inverse of the mask to re-weight the loss map. Similarly, this soft mask tends to approach infinity. A logarithm term is used to regularize it. D3VO [7], a hybrid direct sparse odometry method, also uses the same soft mask as the former. Furthermore, a learnable depth scale consistency mask [22] is proposed to handle moving objects and occlusions. This soft mask is learned by minimizing the depth scale consistency loss. In addition, some indirect approaches utilize optical flow and semantic segmentation tasks to generate soft masks that represent moving objects [23].

Based on the above discussion, binary masks and soft masks represent two different solutions to solve the noisy pixels in dense photometric loss, filtering them and re-weighting them. The problem of binary masks is that only the most extreme noisy pixels can be confidently found. It is hard to identify the boundary between noisy pixels and meaningful pixels. For

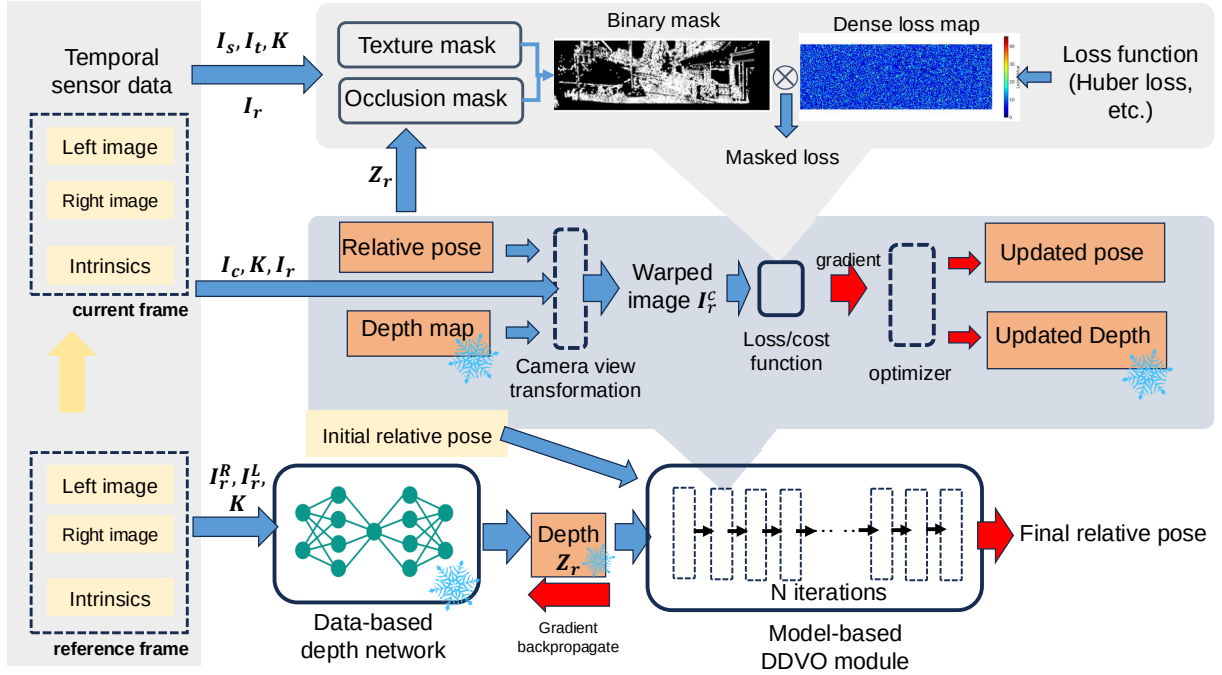


Fig. 2: Structure of hybrid dense direct visual odometry (HDVO). HDVO contains a data-based depth network and a model-based DDVO module. For depth network, the input includes left image  $I_r^R$ , right image  $I_r^L$  of reference frame, and intrinsics  $K$ . DDVO module is an optimization-based method that has  $N$  iterations. For each iteration, relative pose and depth can be updated with an optimizer. Depth network's parameters are optimized with another optimizer. To compute the optimization loss, both the occlusion mask and the texture mask are calculated with the stereo image  $I_s$ , temporal image  $I_t$ , depth map  $Z_r$ , then, the dense loss map is filtered with these binary masks. In the inference stage, the parts with snowflakes  $\star$  are frozen, only the relative pose is optimized efficiently with a second-order optimizer, ESM [18].

soft mask methods, the learning-based method is not stable in training and these masks cannot be generalized to new data. Therefore, we introduce the famous robust loss function, Huber loss function [24], to replace the L1 or L2 function used in precious works [7], [12]. Huber loss function works like L2 function for most loss values, but works like L1 function for high loss values. Therefore, it can decrease the weights of noisy pixels. It is more robust and computation-efficient than the learning-based soft mask. However, for the most extreme noisy pixels, re-weighting by Huber loss can not solve occlusion and low-texture problems. We further propose new binary mask module, including an occlusion mask and a low-texture mask, to filter the extreme noisy pixels. With both Huber loss function and binary mask module, the proposed model can converge to a better result.

The present article builds upon the foundation laid by previously published conference papers [11] [25]. These conference papers presented preliminary findings and initial analyses, which have since been expanded and refined through further research. The first paper proposed to use stereo depth to solve the scale problem of depth map, which can improve direct visual odometry [11]. The second paper proposed a new rule-based mask generation method, which shows better explainability and performance [25]. By integrating and extending the work from these earlier publications, the current article offers a more comprehensive and detailed examination

of the subject matter. This progression allows for a deeper exploration of the topic, addressing limitations and incorporating additional experiments to strengthen the overall conclusions and contributions to the field.

In summary, the contribution of this paper can be summarized as follows.

- This work unifies the modern deep learning network and traditional direct visual odometry method.
- Stereo-based depth is introduced to solve the scale problem of monocular depth map, which is important for generalization in different scenarios and sensors.
- HDVO can realize efficient test-time training, which improves the generalization and keeps real-time inference.
- Robust Huber loss is introduced to replace unstable pixel-wise loss weights, reducing the influence of noisy pixels. The most extreme noisy pixels are filtered using robust binary mask modules.

The paper is structured as follows: Section I provides an introduction to our work, including our motivation, an overview of related works in the field, and a description of our contributions. In Section II, we present HDVO method in detail, outlining its key components and explaining how it addresses the challenges of occlusion, brightness discrepancies, and homogeneous texture. Section III presents the experimental results obtained through the application of HDVO. We analyze and discuss the outcomes of our experiments to evaluate

the effectiveness and performance of the proposed method. Finally, in the concluding section, we summarize the main findings of our study and discuss the potential implications of HDVO in the context of visual odometry. We also highlight avenues for future research and development in this area.

## II. PROPOSED APPROACH

In this section, we first describe the overall framework of HDVO, and then introduce the data-based depth network, model-based DDVO, photometric loss function, and occlusion/texture masks for loss. As shown in Fig. 2, HDVO contains a data-based depth module, a model-based DDVO localization module, and masks for loss. This model is trained with a photometric loss. In the training stage, the initialized pose is set as the ground truth camera pose to optimize the deep learning-based depth network. In the inference stage, the pose is initialized with the pose of the last frame, and the predicted camera pose can be obtained from DDVO module.

### A. Data-based depth module

Dense direct visual odometry is based on RGB-D data. However, the RGB-D data is not available for most scenes, and RGB-D hardware has a higher cost than RGB camera. This section will discuss the stereo solution to generate depth information from RGB images. Compared with the popular monocular depth networks [12], [13], the stereo network can provide a more robust and accurate depth map.

To obtain depth information, the reliable and accurate two-stage 2D-3D stereo network [26], [27], [28] is utilized. The structure of the two-stage 2D-3D stereo network is introduced as follows. It contains 2D feature extraction, 4D cost volume construction, and 3D stereo-matching. Fig. 3 shows this network structure.

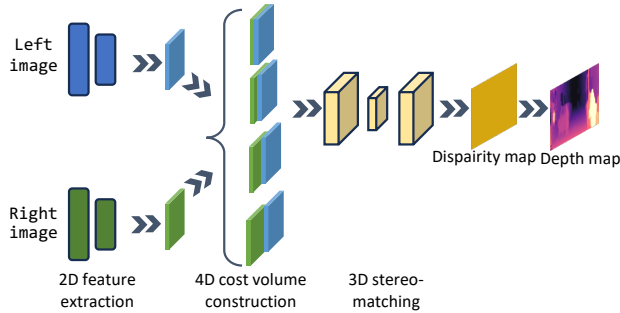


Fig. 3: Structure of two-stage 2D-3D stereo network.

To obtain accurate and robust depth, we propose to use the 3D-CNN (Convolution Neural Network) deep stereo network [26] instead of the monocular auto-encoder [10], [7], [12]. This 3D CNN stereo network contains two stages: a parameter-shared siamese 2D CNN feature extractor and a 3D CNN matching network. Meanwhile, a 4D raw cost volume is built before the matching network. For the 4D cost volume,  $n$  possible stereo matches for the left and right views are enumerated.

Firstly, 2D feature extraction is a shallow 2D CNN to decrease computation costs. This structure is designed similarly

to the residual network [29], [26]. A left feature map and a right feature map will be obtained with this network.

Secondly, similar to the traditional stereo-matching algorithms [30], a 4D cost volume will be built with the left feature map and the right feature map. Assume the disparity distribution of these stereo images ranges from 0 to  $D - 1$ , there are  $D$  stereo-matching pairs to be generated. For each disparity value  $d \in [0, D)$ , there is a homogeneous disparity map  $\mathbf{D}_{v=d}$  only contains value  $d$ , a warped left feature map  $\mathbf{F}_L^R$  from the right feature map is generated with this homogeneous disparity map  $\mathbf{D}_{v=d}$ . Then, a stereo-matching pair is obtained by concatenating the left feature  $\mathbf{F}_L$  and the warped left feature  $\mathbf{F}_L^R$ . This stereo-matching pair is a 3D volume with  $2C \times \frac{1}{4}H \times \frac{1}{4}W$  size. With  $D$  stereo-matching pairs, a 4D cost volume, which has  $D \times 2C \times \frac{1}{4}H \times \frac{1}{4}W$  size, will be obtained.

Thirdly, the 3D stereo-matching is usually performed with stacked 3D auto-encoder CNN as [26]. After this 3D stereo-matching network and  $\times 4$  spatial upsampling, the learned cost volume, which has  $D \times H \times W$  size, can indicate the true disparity value for each pixel. The true disparity value has the highest likelihood along the disparity dimension of the learned cost volume. To make the network differential, a SoftArgmin [31] layer is used. In each  $H \times W$  spatial position, the predicted disparity  $\hat{d}$  is obtained by computing the weighted sum of the cost volume probability vector and disparity distribution vector  $\mathbf{d} = [0, 1, \dots, D - 1]$ .

After having the predicted disparity map  $\hat{\mathbf{D}}$ , the predicted depth map  $\hat{\mathbf{Z}}$  can be transformed from the disparity map  $\hat{\mathbf{D}}$  according to the stereo vision theory, as Eq. 1 shown.

$$\hat{\mathbf{Z}} = \frac{f \cdot b}{\hat{\mathbf{D}}} \quad (1)$$

where  $f$  is the focal length,  $b$  is the baseline length between stereo cameras.

### B. Model-based localization module

The model-based localization module is a traditional dense direct visual odometry (DDVO) which is also an iterative optimization-based method. The DDVO contains a forward image warping and a backward optimization solver. In this section, we introduce the forward image warping. The optimization will be described in the next section. The DDVO is optimized with photometric loss which is computed between the generated image  $\mathbf{I}_r^c$  and ground truth image  $\mathbf{I}_r$ . The forward image warping is to compute this generated image.

To compute the warped reference image  $\mathbf{I}_r^c$  from the current image  $\mathbf{I}_c$ , the transformation coordinate  $\mathbf{p}_c = \mathbf{P}_c$  of the current image should be obtained. Firstly, the coordinate  $\mathbf{p}_r = [u_r; v_r]$  of the reference image is firstly projected to a 3D space  $\mathbf{m}_r = [X_r; Y_r; Z_r]$  as Eq. 2.

$$\begin{aligned} \bar{\mathbf{p}}_r &= [u_r; v_r; 1] \\ \bar{\mathbf{q}}_r &= \mathbf{K}^{-1} \times \bar{\mathbf{p}}_r \\ \mathbf{m}_r &= \mathbf{Z}_r \cdot \bar{\mathbf{q}}_r = [X_r; Y_r; Z_r] \end{aligned} \quad (2)$$

where  $\mathbf{K}$  is the camera intrinsics.  $\mathbf{Z}_r$  is the depth of the reference frame.

Secondly, the 3D points of the reference view are transformed to the current view with the  $4 \times 4$  matrix of relative camera pose  ${}^c\mathbf{T}_r$ , as shown in Eq. 3.

$$\begin{aligned}\bar{\mathbf{m}}_r &= [\mathbf{m}_r; 1] = [X_r; Y_r; Z_r; 1] \\ \bar{\mathbf{m}}_c &= {}^c\mathbf{T}_r \cdot \bar{\mathbf{m}}_r = [X_c; Y_c; Z_c; 1] \\ \mathbf{m}_c &= [X_c; Y_c; Z_c]\end{aligned}\quad (3)$$

Thirdly, the 3D points of the current frame are projected back to 2D space. The obtained coordinates  $\mathbf{p}_c \in \mathbf{P}_c$  will be used for image warping operation.

$$\begin{aligned}\bar{\mathbf{q}}_c &= \mathbf{m}_c / Z_c \\ \bar{\mathbf{p}}_c &= \mathbf{K} \times \bar{\mathbf{q}}_c = [u_c; v_c; 1] \\ \mathbf{p}_c &= [u_c; v_c]\end{aligned}\quad (4)$$

Then, the image warping operation will generate the warped reference image  $\mathbf{I}_r^c$  as Eq. 5. The photometric loss will be computed between  $\mathbf{I}_r^c$  and  $\mathbf{I}_r$ , which will be described in the next section.

$$\mathbf{I}_r^c = \mathbf{W}(\mathbf{I}_c, \mathbf{P}_c) \quad (5)$$

where  $\mathbf{W}()$  is the warping operation according to the target coordinates.

### C. Loss function

The optimization of HDVO is mainly based on the image photometric loss matrix between the generated image and the ground truth image. In the previous works [10], [12], the photometric loss usually refers to the L1/L2 error. In this paper, we use the robust Huber loss as the photometric loss, which is shown in Eq. 6.

$$l_{huber} = \begin{cases} \sum_{\mathbf{p} \in \mathbf{P}} \frac{1}{2} (I_r^c(\mathbf{p}) - I_r(\mathbf{p}))^2 & |I_r^c(\mathbf{p}) - I_r(\mathbf{p})| < \delta \\ \sum_{\mathbf{p} \in \mathbf{P}} \delta (|I_r^c(\mathbf{p}) - I_r(\mathbf{p})| - \frac{1}{2} \delta) & \text{otherwise} \end{cases} \quad (6)$$

The difference between robust loss function and L1/L2 loss function is shown in Fig. 4. Compared with L2 loss, Huber loss can re-weight the outliers. Compared with L1 loss, Huber loss is differentiable anywhere.

In addition, there are depth regularization terms that are applied on the predicted depth  $\hat{\mathbf{Z}}$ . These regularization terms include depth structure similarity error [32], and a depth smoothness loss.

The Structural Similarity Index (SSIM) is a metric used to measure the similarity between two images [32]. In this context, SSIM is used to compare the similarity between generated and ground truth images.

The SSIM loss is shown as follows.

$$l_{structure} = \sum_{\mathbf{p} \in \mathbf{P}} \frac{1}{2} (1 - SSIM(\mathbf{I}_b, \mathbf{I}_b^a)(\mathbf{p})) \quad (7)$$

The details of the SSIM measurement are shown as Eq. 8. Let  $\mu_x$  and  $\mu_y$  be the mean pixel intensity of images  $\mathbf{I}_x, \mathbf{I}_y$ , respectively. Let  $\sigma_x^2$  and  $\sigma_y^2$  be the variances of images  $\mathbf{I}_x, \mathbf{I}_y$ , and let  $\sigma_{xy}$  be the covariance of images  $\mathbf{I}_x, \mathbf{I}_y$ . Additionally, let

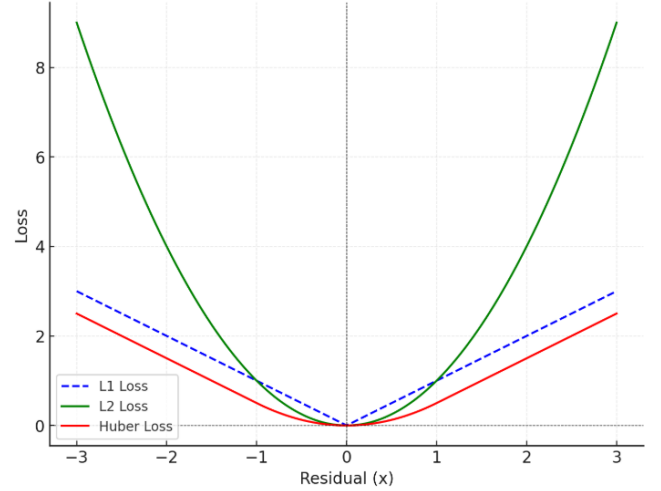


Fig. 4: Different types of robust loss functions.

$c_1$  and  $c_2$  be two variables that help stabilize the division in cases of weak denominator.

$$SSIM(\mathbf{I}_x, \mathbf{I}_y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (8)$$

The disparity smoothness loss can be seen in Eq. 9.

$$l_{smooth} = \sum_{\mathbf{p} \in \mathbf{P}} |\partial_x \hat{\mathbf{Z}}(\mathbf{p})| e^{-|\partial_x I(\mathbf{p})|} + |\partial_y \hat{\mathbf{Z}}(\mathbf{p})| e^{-|\partial_y I(\mathbf{p})|} \quad (9)$$

where  $\partial_x, \partial_y$  refer to the gradient of  $x$  and  $y$  direction of the image separately.  $\hat{\mathbf{Z}}, \mathbf{I}$  are the depth and the image on the same camera view.

### D. Optimization

In this section, we discuss the optimization of HDVO. The optimization of HDVO is described following training and inference stages.

1) *Optimization in training stage:* In training stage, the optimization target variables include data-based network's parameters  $\mathbf{x}_d$  and depth  $\hat{\mathbf{Z}}$  / pose  $\hat{\mathbf{T}}$  variables in DDVO module. There are a large amount of parameters and gradients that requires to be computed. And there are a large number of training data. A first-order optimizer, such as RMSprop, is used in training stage to keep a lower global computation cost in training stage.

Firstly, the gradients of the photometric loss relative to the network parameters are computed as follows.

$$\mathbf{g}_{\mathbf{x}} = \frac{\partial l}{\partial \mathbf{I}_r^c} \cdot \frac{\partial \mathbf{I}_r^c}{\partial \hat{\mathbf{Z}}} \cdot \frac{\partial \hat{\mathbf{Z}}}{\partial \hat{\mathbf{T}}} \cdot \frac{\partial \hat{\mathbf{T}}}{\partial \mathbf{x}_d} \quad (10)$$

where  $l$  is the total loss of photometric loss and depth regularization losses,  $l = l_{huber} + l_{structure} + l_{smooth}$ .

Then, the gradients of the photometric loss relative to depth  $\hat{\mathbf{Z}}$  / pose  $\hat{\mathbf{T}}$  variables in DDVO module are computed as Eq. 11 and Eq. 12.

$$\mathbf{g}_{\hat{\mathbf{Z}}} = \frac{\partial l}{\partial \mathbf{I}_r^c} \cdot \frac{\partial \mathbf{I}_r^c}{\partial \hat{\mathbf{Z}}} \quad (11)$$

$$\mathbf{g}_{\hat{\mathbf{T}}} = \frac{\partial l}{\partial \mathbf{I}_r^c} \cdot \frac{\partial \mathbf{I}_r^c}{\partial \hat{\mathbf{T}}} \quad (12)$$

These parameters are updated with a first-order optimizer as Eq. 13.

$$\Delta \mathbf{x}_d = -\mathbf{g} \quad (13)$$

2) *Optimization in inference stage:* Like most deep learning methods, data-based network do not require to be optimized in inference/test stage. Instead, the optimization goal is transformed to optimize the relative pose  $\hat{\mathbf{T}}$  of moving camera, which is the goal of visual odometry task. The network's parameters and depth variable in DDVO module are frozen.

To be efficient, the DDVO uses the ESM (Efficient Second-order optimization Method) optimizer [18], which is computed as Eq. 14.

There are two stages. At the first stage, the 6D camera pose update  $\Delta \mathbf{x}_p$  is computed.

$$\Delta \mathbf{x}_p = - \left( \left( \frac{\bar{\mathbf{J}} + \hat{\mathbf{J}}}{2} \right)^\top \left( \frac{\bar{\mathbf{J}} + \hat{\mathbf{J}}}{2} \right) \right)^{-1} \left( \frac{\bar{\mathbf{J}} + \hat{\mathbf{J}}}{2} \right)^\top (\mathbf{I}_r^c - \mathbf{I}_r) \quad (14)$$

where the  $\bar{\mathbf{J}}$  is the Jacobian matrix with gradients of the ground truth image,  $\hat{\mathbf{J}}$  is the Jacobian matrix with gradients of the generated image. The Jacobian matrix is computed as follows.

$$\mathbf{J} = \frac{\partial \mathbf{I}}{\partial \mathbf{p}} \cdot \frac{\partial \mathbf{p}}{\partial \Delta \mathbf{x}_p} \quad (15)$$

where  $\mathbf{p} = [u; v]$  is the image coordinate.

The image gradient can be computed easily.

$$\frac{\partial \mathbf{I}_r}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{I}}{\partial u_r} & \frac{\partial \mathbf{I}}{\partial v_r} \end{bmatrix} \quad (16)$$

The gradient of the coordinates relative to 6D parameters is:

$$\frac{\partial \mathbf{p}}{\partial \mathbf{x}_p} = \begin{pmatrix} f_u s(s_0 + s_2) & f_u s(0 - a_2) d_1 \\ f_v s(0 - a_1) d_2 & f_v s(s_0 + s_1) \end{pmatrix} \cdot \begin{pmatrix} Z & 0 & Z \cdot X_w & (0 - Y'_r) X_w & 1 + X'_r X_w & 0 - Y'_r \\ 0 & Z & Z \cdot (0 - Y_w) & -1 - Y'_r Y_w & X'_r Y_w & X'_r \end{pmatrix} \quad (17)$$

where  $[X'_r, Y'_r] = \mathbf{q}$  can be obtained as Eq. 2.  $X_w, Y_w$  is computed as follows.

$$\begin{aligned} X_w &= (X'_r - t_1/Z) \cdot 1/s_0 \\ Y_w &= (Y'_r - t_2/Z) \cdot 1/s_0 \end{aligned} \quad (18)$$

where  $[t_1; t_2; t_3] = \mathbf{t}$  is the translation of the initialized camera pose. The  $s_0, s_1, s_2$  is computed as follows,  $s = 1/(s_0 + s_1 + s_2)$ .

$$\begin{aligned} [s_0; s_1; s_2] &= [1 - \frac{t_3}{Z}; (t_3 X'_r - t_1) f_u \frac{\partial \frac{1}{Z}}{\partial u}; (t_3 Y'_r - t_2) f_v \frac{\partial \frac{1}{Z}}{\partial v}] \\ [a_1; a_2] &= [t_3 X'_r - t_1; t_3 Y'_r - t_2] \\ [d_1; d_2] &= [f_u \frac{\partial \frac{1}{Z}}{\partial u}; f_v \frac{\partial \frac{1}{Z}}{\partial v}] \end{aligned} \quad (19)$$

At the second stage, the 6D update  $\Delta \mathbf{x}_p$  is transformed to  $4 \times 4$  camera pose  $\hat{\mathbf{T}}$ . The parameters  $\mathbf{x}_p \in \mathbb{R}^6$  are defined in

Lie algebra. It is expressed with the constant velocity twist, as the Eq. 20.

$$\Delta \mathbf{x}_p = (\omega \Delta t, \mathbf{v} \Delta t) \quad (20)$$

where  $\mathbf{v}$  is the linear velocity,  $\omega$  is the angular velocity.

Then, incremental camera pose  $\Delta \mathbf{T}(\mathbf{x}_p)$  can be expressed as Eq. 21.

$$\Delta \mathbf{T}(\Delta \mathbf{x}_p) = \exp \left( \begin{bmatrix} [\omega \Delta t]_\times & \mathbf{v} \Delta t \\ 0 & 0 \end{bmatrix} \right) \quad (21)$$

As shown in Eq. 22, the predicted camera pose  $\hat{\mathbf{T}}$  is updated iteratively.

$$\hat{\mathbf{T}}_k = \hat{\mathbf{T}}_{k-1} \Delta \mathbf{T}(\Delta \mathbf{x}_p) \quad (22)$$

It is the product of the last step  $\hat{\mathbf{T}}_{k-1}$  and an incremental pose  $\Delta \mathbf{T}(\Delta \mathbf{x}_p)$ . The  $\mathbf{x}_p$  is a minimal parameter of the Lie algebra  $se(3)$ . In theory, there is an optimal  $\hat{\mathbf{x}}$  to make  $\hat{\mathbf{T}}(\hat{\mathbf{x}}) = \bar{\mathbf{T}}$ .  $\bar{\mathbf{T}}$  is the ground truth camera pose.

### E. Loss masks

As discussed above, HDVO is optimized with dense photometric loss between the generated and original image. According to the properties of cameras, extreme noises can not be avoided in this photometric loss matrix because of the occlusion and homogeneous texture problem. Previously, robust loss functions (e.g. Huber) were used to remove the effect of outliers [24].

However, for photometric loss matrix, those extreme noisy pixels can not be solved well by robust loss function, especially for difficult cases. A binary mask is essential to solve those extreme noisy pixels.

We summarize the extreme noises into two cases: occlusion and low-texture. The occlusion mask removes all noises from view occlusion and dynamic objects, and only keeps the most reliable losses. The texture mask removes the ambiguous pixels which can not be discriminated from surrounding pixels.

1) *Occlusion mask:* For a stereo direct visual odometry model, there are three kinds of occlusion, including stereo view occlusion and temporal view occlusion. The optimization goal is to let loss  $l$  be 0, as shown in Eq. 23, but the occlusion problem makes it impossible.

$$l(\mathbf{p}) = |\hat{l}(\mathbf{p}) - \bar{l}(\mathbf{p})| = \begin{cases} = 0, \mathbf{p} \in \text{nonocclusion} \\ > \gamma, \mathbf{p} \in \text{occlusion} \end{cases} \quad (23)$$

where  $\mathbf{p} = [u; v]$  is the pixel coordinate. For the occlusion pixels, the photometric loss can not be zero because there is no matched pixel. There is always a minimum error  $\gamma$ , which is the optimization noise. Therefore, the occlusion pixels will affect the model converging to a global minimum.

To keep a reliable loss, a new method is proposed to find the occlusion, namely *stereo-temporal consistency (STC) occlusion mask*. This method is based on the double-check of stereo-warped result and temporal-warped result on the same camera view. The pixels that can be recovered from both stereo and temporal warping are regarded as the confident non-occlusion pixels. The computation steps of the STC occlusion mask are shown in Fig. 5.

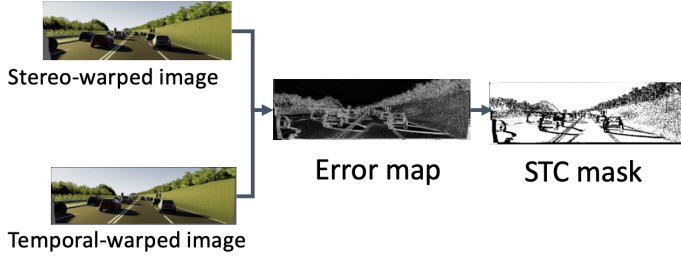


Fig. 5: Steps for computing the STC occlusion mask.

Firstly, the stereo warped stereo image  $\mathbf{I}_r^s$  and the temporal warped temporal image  $\mathbf{I}_r^t$  are obtained.

$$\begin{aligned}\mathbf{I}_r^s &= \mathbf{W}(\mathbf{I}_s, \mathbf{Z}_r) \\ \mathbf{I}_r^t &= \mathbf{W}(\mathbf{I}_t, \mathbf{Z}_r)\end{aligned}\quad (24)$$

where  $\mathbf{I}_r^s$  and  $\mathbf{I}_r^t$  are warped images in the same reference view.  $\mathbf{I}_s$  and  $\mathbf{I}_t$  are the stereo image and adjacent temporal image.  $\mathbf{Z}_r$  is the reference depth. Then, the error map  $\mathbf{E}_r$  between  $\mathbf{I}_r^s$  and  $\mathbf{I}_r^t$  is computed. To generate a binary mask for the loss map, setting a suitable threshold value on this error map yields the STC binary mask  $\mathbf{M}_{STC}$  as Eq. 25.

$$\mathbf{M}_{STC} = \begin{cases} 1, \mathbf{E}_{err} < \delta \\ 0, \mathbf{E}_{err} \geq \delta \end{cases}\quad (25)$$

In addition, different camera views have brightness discrepancies problems, which are extra noises when measuring the error in Fig. 5. To solve this problem, a brightness robust error is proposed. To keep this error robust for the brightness discrepancy of different camera views, we model this error with the image's local patches instead of the single pixel. Furthermore, instead of simply summing or averaging the error of the local patch, we introduce the zero mean normalized cross-correlation  $c$  to model the error of the prediction and ground truth image local patches, as shown in Eq. 26:

$$c(\mathbf{p}_g) = \frac{1}{\sum \mathbf{P}_l} \sum_{\mathbf{p} \in \mathbf{P}_l} \frac{(\hat{I}(\mathbf{p}) - \hat{\mu}) \cdot (\bar{I}(\mathbf{p}) - \bar{\mu})}{\hat{\sigma} \bar{\sigma}}\quad (26)$$

where  $\hat{\times}, \bar{\times}$  refer to the prediction and the ground truth.  $\mu$  is the average value of the local image patch.  $\sigma = \sum_{\mathbf{p} \in \mathbf{P}_l} \sqrt{(I(\mathbf{p}) - \mu)^2}$  is the standard deviation of the local image patch.  $\mathbf{P}_l$  is the coordinates of the local image patch.  $\mathbf{p}_g$  is the position in the global image, which is the center of  $\mathbf{P}_l$ .

Then, we formulate the brightness robustness (BR) error as Eq. 27.

$$E_r(\mathbf{p}) = \frac{1}{\sum \mathbf{P}_g} \sum_{\mathbf{p} \in \mathbf{P}_g} (1 - c(\mathbf{p}))\quad (27)$$

where  $\mathbf{P}_g$  is the global image coordinates. As the image intensities are normalized by the average and the standard deviation, the BR error has the range of  $[0, 2]$ .

2) *Texture mask*: For the homogeneous texture or low-texture area, the photometric loss will fail because there is the same pixel intensity in a local area. Intuitively, humans also can not estimate the correct depth from the homogeneous texture area. In this part, a simple but efficient texture mask is proposed.

As shown in Eq. 28, if the optimization converges to the optimal, the prediction and the ground truth have the same position  $\mathbf{p}$ . However, if a pixel belongs to a homogeneous texture area, the loss becomes zero even if the prediction position  $\hat{\mathbf{p}}$  is not equal to the ground truth position  $\bar{\mathbf{p}}$ . This kind of noise will affect the optimization of photometric loss. The model can not converge to the minimum point.

$$l(\mathbf{p}) = |\hat{I}_r(\hat{\mathbf{p}}) - \bar{I}_r(\bar{\mathbf{p}})| \begin{cases} = 0, \hat{\mathbf{p}} = \bar{\mathbf{p}} \\ = 0, \hat{\mathbf{p}} \neq \bar{\mathbf{p}} \text{ and } |\hat{\mathbf{p}} - \bar{\mathbf{p}}| < R \end{cases}\quad (28)$$

Homogeneous texture area refers to the local image patch with the same intensity. If the local image patch is small enough, we can assume a new hypothesis: a pixel has the same intensity as its surrounding local image patch. It is a homogeneous texture pixel.

A new texture mask is proposed, namely local average max (LAM) texture mask. The computation steps of the LAM occlusion mask are shown in Fig. 6.

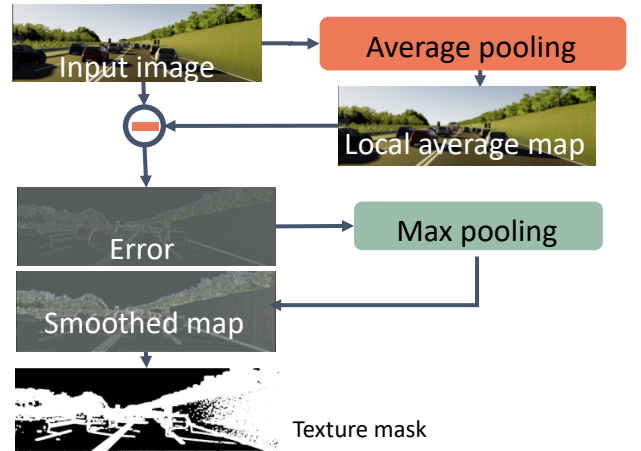


Fig. 6: Steps for computing the LAM texture mask.

Firstly, the LAM mask models the local patches of each position  $\mathbf{p}$  using the average intensities of the local patches. To realize that, as shown in Eq. 29, we first compute the global average map using the average pooling operation. The kernel size is the same as the local patch size, e.g. (5,5). The stride is 1. The average pooling can realize this step efficiently.

$$I_{avg}(\mathbf{p}_g) = \frac{1}{\sum \mathbf{P}_l} \sum_{\mathbf{p} \in \mathbf{P}_l} I(\mathbf{p})\quad (29)$$

where  $I_{avg}(\mathbf{p})$  is the average value of the image local patch  $\mathbf{P}_l$  which has the center  $\mathbf{p}_g$ ,  $\mathbf{p}_g \in \mathbf{P}_g$ .  $\mathbf{P}_g$  is the coordinates of the global image.

Then, L1 intensity error between the original image  $I$  and the average map  $I_{avg}$  is computed as Eq. 30.

$$E_{err}(\mathbf{p}_g) = |I(\mathbf{p}_g) - I_{avg}(\mathbf{p}_g)| \quad (30)$$

However, if we directly generate the binary mask from this error map, the binary mask will be noisy. The limited context of the average pooling layer causes this problem. To solve that, we first repeat the average pooling to be two layers to enlarge the context field of the local image patch. Meanwhile, we use a max pooling layer (kernel size = local patch size, stride=1) to filter the outliers in the error map as Eq. 31. The max pooling layer can make the LAM mask to be continuous in the homogeneous texture area.

$$E_{smooth}(\mathbf{p}_g) = \max_{\mathbf{p} \in \mathbf{P}_l} E_{err}(\mathbf{p}) \quad (31)$$

where  $E_{smooth}(\mathbf{p}_g)$  is the max pooling output of one local patch.

Finally, the LAM binary mask is obtained with a threshold  $\beta$  on the error map  $E_{smooth}$ :

$$\mathbf{M}_{LAM} = \begin{cases} 0, & E_{smooth} < \beta \\ 1, & E_{smooth} \geq \beta \end{cases} \quad (32)$$

If the error of pixel  $\mathbf{p}_g$  is lower than the threshold, it will be regarded as a homogeneous texture pixel.

### III. EXPERIMENTS

In the experiment part, the new method, HDVO, is first compared with the previous state-of-the-art methods, including traditional SLAM, deep learning networks, as well as hybrid visual odometry methods. Then, HDVO is analyzed from different aspects: loss masks, loss function, depth quality, computation cost, and diverse datasets.

#### A. Datasets

Five datasets are used for the abundant experiments in this paper. There are three datasets for the ground vehicle: KITTI Odometry, KITTI Depth, and Virtual KITTI2, and two datasets for the drone: MidAir and EuRoC MAV.

- 1) KITTI Odometry: Nine sequences (00-08) are used for training, and two sequences (09, 10) are used for testing, as in most papers. For the depth estimation network, the network is optimized with 5 training epochs on this dataset.
- 2) KITTI Depth: Similarly to the previous work [12], we re-organize this dataset by locating the possible  $\pm 1$  temporal adjacent images in the raw data, allowing us to perform temporal image warping and reconstruction. Following previous works, the Eigen split and evaluation methods are used.
- 3) Virtual KITTI2: This is a common virtual dataset similar to the KITTI dataset. Scene{01,02,06,18} are used for the training of the depth estimation network, Scene20 is used for the evaluation. Only the data with ‘morning’ weather is used. The depth estimation network is trained over 30 epochs.
- 4) MidAir: This is a drone simulation dataset. We use all 30 sequences of ‘sunny’ weather for the training of the depth network, and 3 VO testing sequences are

used for evaluation. Because the test sequences contain 14990, 14990, 7867 frames, and the trajectories are repeated patterns, we only keep the first 1499, 1499, 787 frames for the evaluation.

- 5) EuRoC MAV: This is also a widely used drone dataset. Following previous works, *MH\_03\_m* *MH\_05\_d* *V1\_03\_d* *V2\_02\_m* are used for the evaluation.

#### B. Details

For the training of the depth estimation network, we use an RMSprop optimizer with  $1e-4$  learning rate. The learning rate follows the Cosine Annealing updating policy, with a minimum learning rate of  $1e-6$ . A liner warmup method is also used at the first 100 iterations. In addition, we use L2 norm gradient clip during the training with a maximum norm of 35. For most experiments, we optimize the network for 5 epochs with a batch size of 3 on three Nvidia A40 GPUs.

For the setting of the visual odometry part, we use the gray-scale sequential images of the left camera. The implementation of the direct visual odometry refers to the OpenROX code [33]. The masking strategy is used for visual odometry. We also use the robust Huber loss to get better odometry results in part of experiments.

#### C. Experiments of comparison with state-of-the-art methods

| Method              | seq.9       |           | seq.10      |           |
|---------------------|-------------|-----------|-------------|-----------|
|                     | $t_{err}$   | $r_{err}$ | $t_{err}$   | $r_{err}$ |
| ORBSLAM [1]         | 15.30       | 0.26      | 3.68        | 0.48      |
| ORBSLAM2-mono [2]   | 57.60       | 58.03     | 62.37       | 57.68     |
| ORBSLAM2-stereo [2] | 0.85        | 0.26      | 0.56        | 0.24      |
| DSO-stereo [34]     | 41.04       | 14.47     | 1.34        | 0.42      |
| DSO-mono [4]        | 74.81       | 41.27     | 94.88       | 39.04     |
| SfmLearner [13]     | 17.84       | 6.78      | 37.91       | 17.80     |
| Geonet [35]         | 43.76       | 16.00     | 35.60       | 13.8      |
| DeepVO [36]         | -           | -         | 5.96        | 6.12      |
| Wang [37]           | 9.30        | 3.50      | 7.21        | 3.90      |
| Li [38]             | 8.10        | 2.81      | 12.90       | 3.17      |
| TrianFlow [39]      | 6.93        | 0.44      | 4.66        | 0.62      |
| TAPE [40]           | 6.72        | 2.60      | 8.66        | 3.13      |
| F2FPE [40]          | 2.36        | 1.06      | 3.00        | 1.28      |
| AdaMem              | -           | -         | 3.94        | 1.73      |
| TartanVO [41]       | 6.00        | 3.11      | 6.89        | 2.73      |
| MaskVO [42]         | 8.51        | 2.79      | 9.74        | 3.34      |
| DeepAVO [43]        | -           | -         | 5.49        | 2.49      |
| LoSeVO [44]         | -           | -         | 5.59        | 2.66      |
| CoProU-VO [45]      | 4.70        | 1.89      | 7.64        | 3.67      |
| DVSO [8]            | 0.83        | 0.21      | 0.74        | 0.21      |
| UnOS [9]            | 5.21        | 1.80      | 5.20        | 2.18      |
| D3VO [7]            | 0.78        | $\times$  | 0.62        | $\times$  |
| DFVO [10]           | 2.07        | 0.23      | 2.06        | 0.36      |
| DDSO [46]           | 15.02       | 0.20      | 8.76        | 0.66      |
| DINO-VO [47]        | 1.64        | 0.93      | 0.96        | 0.71      |
| DPV-SLAM++ [48]     | 17.69       | 0.23      | 6.32        | 0.23      |
| HDVO [11]           | 0.87        | 0.28      | 0.87        | 0.46      |
| Ours                | <b>0.73</b> | 0.36      | <b>0.42</b> | 0.24      |

TABLE I: Results with with KITTI metrics  $t_{err}, r_{err}$

Firstly, we show the comparison of the proposed HDVO with other state-of-the-art methods. We report the comparison of the results in Tab. I and Tab. II. KITTI sequence error metric and absolute trajectory error are used to evaluate odometry results. HDVO has achieved better performance than other

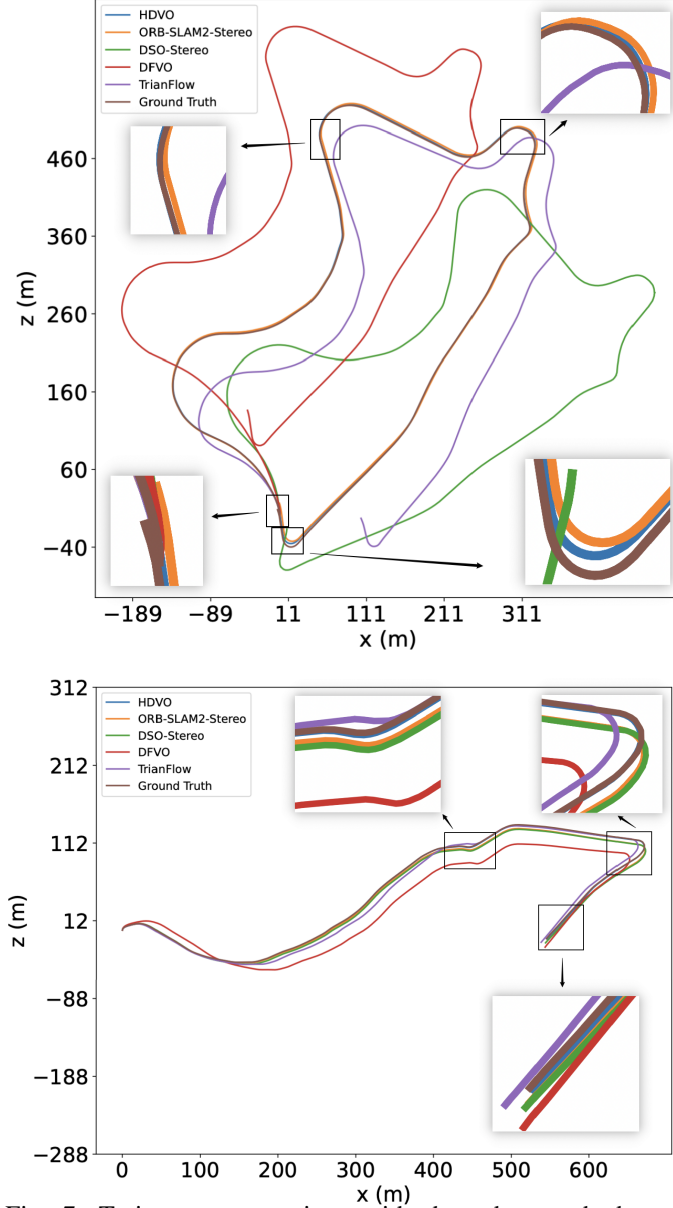


Fig. 7: Trajectory comparison with the other methods on KITTI sequence 09 and 10.

| Method          | seq.9                | seq.10               |
|-----------------|----------------------|----------------------|
|                 | <i>ATE</i>           | <i>ATE</i>           |
| ORB full [1]    | 0.0140±0.0080        | 0.0120±0.0110        |
| ORB short [1]   | 0.0640±0.1410        | 0.0640±0.1300        |
| SfmLearner [13] | 0.0160±0.0090        | 0.0130±0.0090        |
| Geonet [35]     | 0.0120±0.0070        | 0.0120±0.0090        |
| Vid2depth [17]  | 0.0130±0.0100        | 0.0120±0.0110        |
| Com Col [23]    | 0.0120±0.0070        | 0.0120±0.0080        |
| UnOS [9]        | 0.0120±0.0060        | 0.0130±0.0080        |
| HDVO [11]       | 0.0109±0.0068        | 0.0105±0.0088        |
| Ours            | <b>0.0109±0.0064</b> | <b>0.0099±0.0089</b> |

TABLE II: Results with *ATE* metric on KITTI seq 09, 10.

visual odometry methods. Compared to previous methods, the new solution achieves the smallest translation error and competitive rotation error results.

In addition, the comparisons of inference speed and odometry errors (*ATE*, translation error, rotation error) are shown in Fig. 8. HDVO has a significant advantage in terms of both inference speed and odometry error. These results are measured on the same cluster node with four 2080Ti GPUs. HDVO can run with less than 22 ms. Another advantage of HDVO is that it can recover the scale of the real world. Fig. 8 suggests that monocular-based visual odometry methods can not predict the correct scale of the real world. Some results in Tab. I are re-scaled using ground truth odometry results, which are reported in original papers [10], [7], [8], [46], [9], [39]. Therefore, all previous hybrid visual odometry methods can not be directly used in real autonomous driving localization.

Finally, the comparison of HDVO and some famous visual odometry methods is shown in Fig. 7. These trajectories are re-scaled using ground truth odometry results because several methods can not predict the odometry results with the correct scale of the real world. Based on these two figures, HDVO is the best one to fit the ground truth trajectory. Based on the detailed visualization images, ORB-SLAM2 still has large shifts in several areas, although it has a close odometry error as HDVO.

#### D. Ablation study for Loss function

This experiment suggests the advantage and importance of robust loss function in HDVO optimization. In this experiment, a simplified PSMNet is used to predict depth. The quality of depth maps and camera poses is shown as Tab. IV and Tab. III, which compares robust Huber loss and L1 loss.

| Loss        | Camera Pose Error Metric |                         |                |                  |                   |
|-------------|--------------------------|-------------------------|----------------|------------------|-------------------|
|             | $t_{err}(\%)$            | $r_{err}$<br>(deg/100m) | <i>ATE</i> (m) | $RPE_{tran}$ (m) | $RPE_{rot}$ (deg) |
| seq09-HUBER | <b>0.83</b>              | <b>0.30</b>             | <b>7.08</b>    | <b>0.013</b>     | <b>0.040</b>      |
| seq09-L1    | 1.03                     | 0.40                    | 8.25           | 0.014            | 0.041             |
| seq10-HUBER | <b>1.42</b>              | <b>0.44</b>             | <b>12.25</b>   | <b>0.016</b>     | <b>0.037</b>      |
| seq10-L1    | 1.74                     | 0.52                    | 14.76          | 0.019            | 0.038             |

TABLE III: Camera pose results.

| Loss        | Error Metric |               |              |              | Accuracy Metric |                   |                   |
|-------------|--------------|---------------|--------------|--------------|-----------------|-------------------|-------------------|
|             | abs rel      | rel sqr       | rmse         | rmse log     | 1.25            | 1.25 <sup>2</sup> | 1.25 <sup>3</sup> |
| seq09-HUBER | <b>0.342</b> | <b>12.840</b> | <b>8.466</b> | <b>0.407</b> | <b>86.5</b>     | <b>90.6</b>       | <b>93.1</b>       |
| seq09-L1    | 0.481        | 20.636        | 11.229       | 0.511        | 81.5            | 86.8              | 90.3              |
| seq10-HUBER | <b>0.211</b> | <b>7.941</b>  | <b>7.028</b> | <b>0.307</b> | <b>90.5</b>     | <b>93.9</b>       | <b>95.7</b>       |
| seq10-L1    | 0.369        | 16.208        | 10.426       | 0.434        | 85.7            | 90.1              | 92.8              |

TABLE IV: Depth results with robust loss and L1 loss.

We evaluate differences between Huber loss and L1 loss with depth error, depth accuracy and camera pose error. The above results suggest that robust loss has a significant improvement on the hybrid dense direct visual odometry.

#### E. Analysis of the quality of predicted depth

To validate the effectiveness of data-based depth network, we analyze the depth quality by computing the error and

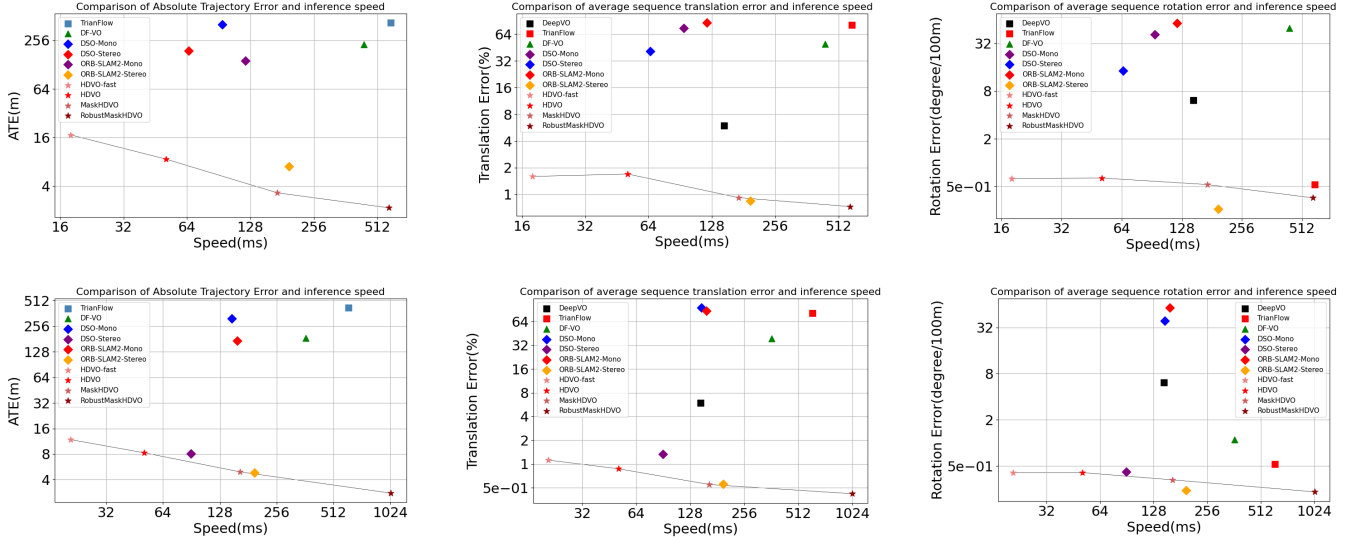


Fig. 8: Comparison of the speed and localization error with the other methods on KITTI sequence 09 (row 1) and 10 (row 2).

accuracy of the predictions. Error and accuracy metrics are described in the previous paper [49]. In this part, we compare the stereo depths results and monocular depths results in the previous visual odometry methods [10], [7], [12], [50], [13], [8], [35]. As shown in Tab. V, we report the results with depth error and depth accuracy metrics.

On almost all metrics, the proposed method achieves the best performance. These results suggest that the deep stereo network can provide more robust and accurate depth prior than the previous solutions. And monocular depth results are re-scaled with ground truth depth, which is not practical. In contrast, our results have the correct scale of the real world.

| Method          | Error Metric |              |              |              | Accuracy Metric |                   |                   |
|-----------------|--------------|--------------|--------------|--------------|-----------------|-------------------|-------------------|
|                 | abs rel      | rel sqr      | rmse         | rmse log     | 1.25            | 1.25 <sup>2</sup> | 1.25 <sup>3</sup> |
| SFMLearner [13] | 0.208        | 1.768        | 6.856        | 0.283        | 67.8            | 88.5              | 95.7              |
| UndeepVO [50]   | 0.183        | 1.730        | 6.570        | 0.268        | <b>X</b>        | <b>X</b>          | <b>X</b>          |
| DVSO[8]         | 0.097        | 0.734        | 4.442        | 0.187        | 88.8            | 95.8              | <b>98.0</b>       |
| GeoNet[35]      | 0.149        | 1.060        | 5.567        | 0.226        | 79.6            | 93.5              | 97.5              |
| MonoDepth2[12]  | 0.106        | 0.806        | 4.630        | 0.193        | 87.6            | 95.8              | <b>98.0</b>       |
| D3VO [7]        | 0.099        | 0.763        | 4.485        | 0.185        | 88.5            | 95.8              | 97.9              |
| HDVO [11]       | 0.080        | 0.795        | 4.146        | 0.185        | 92.2            | 95.9              | 97.6              |
| Masked HDVO     | <b>0.077</b> | <b>0.676</b> | <b>3.863</b> | <b>0.173</b> | <b>92.7</b>     | <b>96.3</b>       | 97.9              |

TABLE V: Depth estimation results compared with monocular networks.

#### F. Computation cost analysis

To further evaluate the efficiency and practical applicability of the proposed HDVO framework, we analyze its inference latency and throughput on different hardware platforms. Specifically, experiments are conducted on two high-performance GPUs (A40 and L40) and two embedded edge devices (NVIDIA Jetson Orin and Jetson Thor). All results are summarized in Tab. VI.

As shown in this table, HDVO achieves real-time inference across all tested platforms while maintaining consistent lo-

| Platform    | Inference Performance |             | Pose Error Metrics |                    |         |
|-------------|-----------------------|-------------|--------------------|--------------------|---------|
|             | Time ↓ (ms)           | FPS ↑       | $t_{err}$ (%)      | $r_{err}$ (°/100m) | ATE (m) |
| A40 GPU     | <b>21.06</b>          | <b>47.5</b> | 1.12               | 0.41               | 11.83   |
| L40 GPU     | 26.67                 | 37.5        |                    |                    |         |
| Jetson Orin | 80.89                 | 12.4        |                    |                    |         |
| Jetson Thor | 55.97                 | 17.9        |                    |                    |         |

TABLE VI: Inference time analysis of HDVO (CoEx backbone [51]).

calization accuracy. On a mid-trim GPU such as the A40, the proposed method reaches an inference speed of up to 47.5 FPS. Even on embedded platforms, HDVO remains computationally efficient, achieving 12.4 FPS on Jetson Orin and 17.9 FPS on Jetson Thor. These results indicate that the proposed method is suitable for real-time visual odometry applications, including those deployed on resource-constrained platforms.

All inference times are measured using the original PyTorch implementation without additional engineering optimizations. Further acceleration is possible through deployment-oriented frameworks such as ONNX or TensorRT. Overall, these results highlight the favorable efficiency–accuracy trade-off of HDVO and its potential for real-world deployment.

#### G. Ablation study for loss masks

1) *Compare with the other masks:* To demonstrate the advantage of the proposed masking method, we compare with several typical masking methods, including principled mask [17], monodepth2 mask [12], SFMLearner mask [13], Bayesian mask [21]. As shown in Tab. VII of the next page, our masking method has the best results on the predicted depth and camera poses.

2) *Effect of proposed masks:* To demonstrate the advantage of the multi-mask system, we show both the depth estimation and visual odometry results on both real and virtual data.

In our experiments, the STC mask and the LAM mask both improve depth prediction accuracy and camera pose

| Type                     | Depth Error   |               |               |               | Depth Accuracy |                   |                   | Pose Error  |             |              |              | Time/ms |
|--------------------------|---------------|---------------|---------------|---------------|----------------|-------------------|-------------------|-------------|-------------|--------------|--------------|---------|
|                          | abs rel       | rel sqr       | rmse          | rmse log      | 1.25           | 1.25 <sup>2</sup> | 1.25 <sup>3</sup> | $t_{err}$   | $r_{err}$   | $RPE_{tran}$ | $RPE_{rot}$  |         |
| <i>Soft</i>              |               |               |               |               |                |                   |                   |             |             |              |              |         |
| SFMLearner [13]          | 0.1482        | 2.7122        | 5.0126        | 0.2497        | 89.59          | 93.96             | 96.10             | 2.32        | 0.66        | 0.024        | 0.039        | 137.6   |
| Bayesian [52], [21], [7] | 0.3186        | 12.3994       | 9.1111        | 0.3943        | 86.29          | 90.91             | 93.40             | 2.64        | 0.70        | 0.024        | 0.039        | 154.6   |
| <i>Binary</i>            |               |               |               |               |                |                   |                   |             |             |              |              |         |
| Principled [17]          | 0.3431        | 14.2719       | 9.7451        | 0.4090        | 85.83          | 90.46             | 93.06             | 2.90        | 0.74        | 0.023        | 0.039        | 209.8   |
| monodepth2 [12]          | 0.0781        | 0.5958        | 3.6738        | 0.1747        | 91.67          | 95.95             | 97.76             | 5.17        | 1.66        | 0.026        | 0.045        | 121.7   |
| Ours                     | <b>0.0665</b> | <b>0.4731</b> | <b>3.3050</b> | <b>0.1460</b> | <b>93.37</b>   | <b>97.20</b>      | <b>98.65</b>      | <b>1.54</b> | <b>0.45</b> | <b>0.021</b> | <b>0.032</b> | 209.6   |

TABLE VII: Compare with the other masking strategies on the depth results on KITTI Odometry sequence 09. For those masks from optical flow [9], [20] are not considered.

| Data Type     | Mask type  | Depth Error Metrics |               |                |               | Depth Accuracy Metric |              |              | Camera Pose Error Metric |                         |                  |                   |
|---------------|------------|---------------------|---------------|----------------|---------------|-----------------------|--------------|--------------|--------------------------|-------------------------|------------------|-------------------|
|               |            | abs rel             | rel sq        | rmse           | rmse log      | $\tau < 1.25$         | $< 1.25^2$   | $< 1.25^3$   | $t_{err}(\%)$            | $r_{err}$<br>(deg/100m) | $RPE_{tran}$ (m) | $RPE_{rot}$ (deg) |
| Real seq09    | Baseline   | 0.2827              | 10.4134       | 8.414          | 0.369         | 86.89                 | 91.53        | 94.03        | 2.77                     | 0.68                    | 0.024            | 0.039             |
|               | LAM mask   | 0.2535              | 8.2864        | 7.538          | 0.347         | 86.91                 | 91.69        | 94.26        | 2.41                     | 0.71                    | 0.024            | 0.037             |
|               | STC mask   | 0.1042              | 1.2005        | 3.934          | 0.200         | 91.29                 | 95.43        | 97.33        | 2.68                     | 0.81                    | 0.024            | 0.038             |
|               | multi-mask | <b>0.0665</b>       | <b>0.4731</b> | <b>3.305</b>   | <b>0.146</b>  | <b>93.37</b>          | <b>97.20</b> | <b>98.65</b> | <b>1.54</b>              | <b>0.45</b>             | <b>0.021</b>     | <b>0.032</b>      |
| Real seq10    | Baseline   | 0.4019              | 14.8759       | 9.528          | 0.456         | 82.49                 | 88.11        | 91.38        | 1.89                     | 0.56                    | 0.016            | 0.042             |
|               | LAM mask   | 0.3388              | 11.1628       | 8.387          | 0.418         | 83.18                 | 88.80        | 92.13        | 1.52                     | 0.47                    | 0.016            | 0.040             |
|               | STC occlu  | 0.1587              | 1.9037        | 4.051          | 0.273         | 87.23                 | 92.33        | 95.13        | 1.66                     | 0.71                    | 0.016            | 0.040             |
|               | multi-mask | <b>0.0840</b>       | <b>0.4262</b> | <b>2.804</b>   | <b>0.175</b>  | <b>91.13</b>          | <b>95.74</b> | <b>97.91</b> | <b>1.48</b>              | <b>0.32</b>             | <b>0.015</b>     | <b>0.038</b>      |
| Virtual seq20 | Baseline   | 0.7648              | 84.2296       | 62.1419        | 0.6397        | 78.75                 | 84.38        | 87.56        | 13.90                    | 3.75                    | 0.066            | 0.061             |
|               | LAM mask   | 0.3273              | 34.7354       | 63.2385        | 0.5206        | 79.81                 | 86.98        | 90.35        | 5.29                     | 0.87                    | 0.045            | 0.017             |
|               | STC mask   | 0.3298              | 19.9154       | 64.7536        | 0.4740        | 79.20                 | 86.60        | 90.29        | 1.68                     | 0.76                    | 0.006            | 0.014             |
|               | multi-mask | <b>0.1284</b>       | <b>8.8084</b> | <b>59.4475</b> | <b>0.3545</b> | <b>83.12</b>          | <b>91.16</b> | <b>94.67</b> | <b>0.95</b>              | <b>0.46</b>             | <b>0.005</b>     | <b>0.013</b>      |

TABLE VIII: Experiments to show the effect of the proposed multiple masks. Three groups of results on both real-world data and virtual data are recorded, and all show significant improvements with the masks.

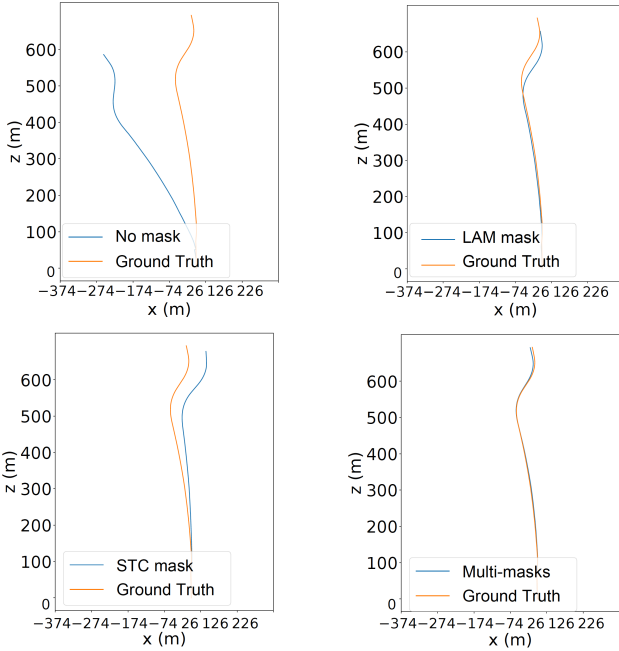


Fig. 9: Comparison of the estimated and actual visual odometry trajectory. The results of four different conditions (no mask, LAM mask, STC mask, and multi-mask) are shown. A  $x-z$  dimension view (bird's eye view) is shown.

prediction accuracy. The improvement is more significant in

the simulation data, Virtual KITTI2. The STC mask has a better effect than the LAM mask, which also suggests that removing the occlusion areas is more important.

The multi-mask system is obtained by combining the STC mask and the LAM mask. For these three sequences, the multi-mask system has the best result. The results on the depth estimation task and the visual odometry task all have significant improvements. These results not only suggest that both the occlusion areas and homogeneous texture areas are important for the computation of the image warping loss, but also demonstrate the effect of the proposed multi-mask system. The above experiment data is reported in Tab. VIII of the next page.

Besides the data in Tab. VIII, we also show the visualization of the estimated trajectory of the visual odometry. Fig. 9 shows the comparison of the localization results with different masking strategies. This visual comparison clearly suggests that the proposed masking strategy is efficient and essential for visual odometry. Finally, the different masks are also shown in Fig. 10 of the next page. Homogeneous texture mask focuses on the low-texture area, such as sky. Occlusion mask focuses on the edge of objects.

#### H. Results on diverse datasets

To show more results on diverse datasets of the proposed method, two visualization results on drone datasets, Mid-Air and EuRoC MAV, are shown in Fig. 11 and Fig. 12. Mid-Air dataset is a drone dataset with far distance. The maximum depth can be over 1000 meters. EuRoC MAV dataset is a drone

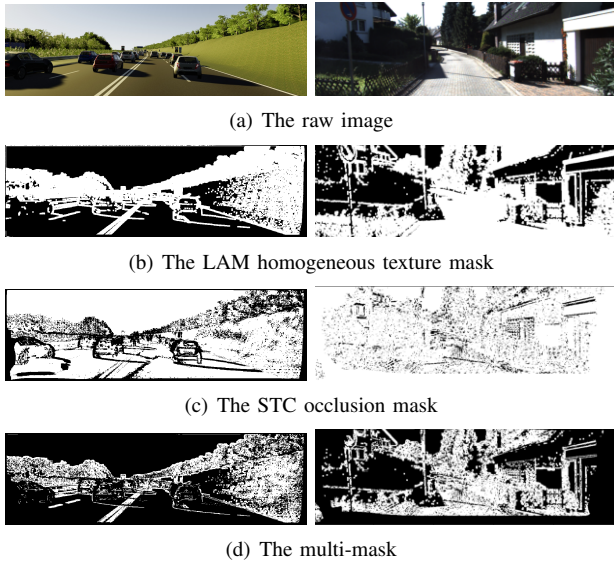


Fig. 10: Masks examples from sequence 20 of the Virtual KITTI2 dataset and sequence 09 and 10 of the KITTI Odometry dataset (from the left to the right). The black areas are the masked pixels. The LAM homogeneous texture mask removes homogeneous texture areas, e.g. the sky, highlight and shadow areas. The STC occlusion mask removes occlusion areas. The multi-mask system is obtained by removing all masked pixels from these masks.

dataset with fast motion. The stereo baseline is also small. These two datasets show different scenarios compared with autonomous vehicle. With the proposed methods, the predicted trajectory can match the ground truth trajectory well.

Then, the ATE metric results suggest that the proposed method can also improve visual odometry accuracy for drone-captured videos.

Tab. IX and Tab. X show the visual odometry results on the EuRoC MAV and MidAir drone datasets separately.

| Seq ID      | <i>MH_03</i>  | <i>MH_05</i>  | <i>V1_03</i>  | <i>V2_02</i>  | Mean          |
|-------------|---------------|---------------|---------------|---------------|---------------|
| HDVO        | 0.0061        | 0.0049        | 0.0095        | 0.0093        | 0.0075        |
| Masked HDVO | <b>0.0052</b> | <b>0.0029</b> | <b>0.0049</b> | <b>0.0078</b> | <b>0.0052</b> |

TABLE IX: Odometry results with ATE metric on the EuRoC MAV dataset.

| Seq ID      | Sunny00       | Sunny01       | Sunny02       | Mean          |
|-------------|---------------|---------------|---------------|---------------|
| HDVO        | 0.1069        | 0.1375        | 0.0757        | 0.1067        |
| Masked HDVO | <b>0.0082</b> | <b>0.0164</b> | <b>0.0073</b> | <b>0.0319</b> |

TABLE X: Odometry results with ATE metric on the Mid-Air dataset.

In the MidAir dataset, some frames have difficult occlusion problems when the drone is close to the forest or the hill in the MidAir dataset. And there are larger homogeneous texture areas (e.g. sky, lake) in this dataset. In EuRoC MAV dataset, there are also many homogeneous texture areas, and the occlusion problem is obvious for indoor scenarios.

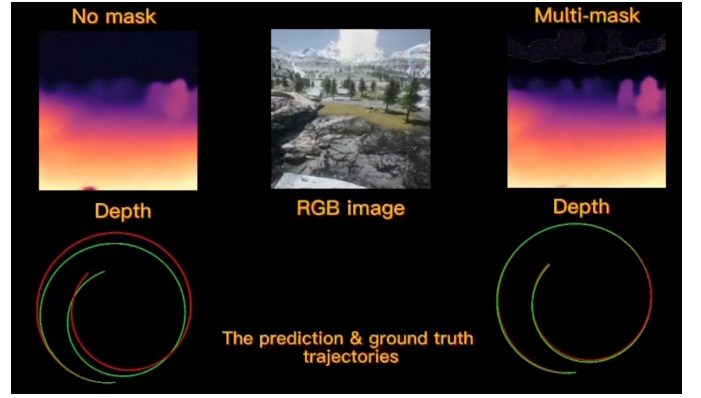


Fig. 11: Results on Mid-Air drone dataset. Red line is the ground truth trajectory, green line is the predicted trajectory. Multi-mask refers to use proposed mask generation module.

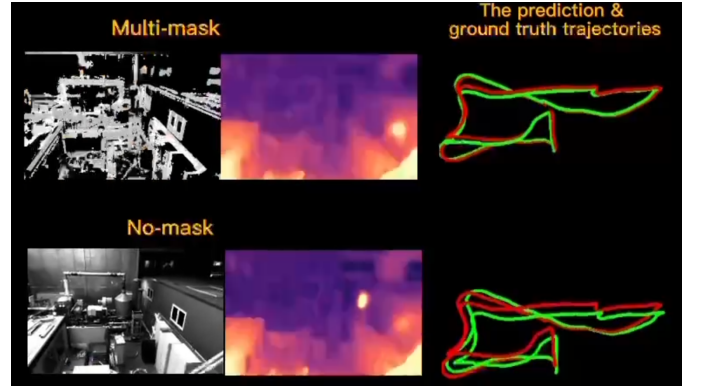


Fig. 12: Results on EuRoC MAV drone dataset.

Therefore, the improvement with the proposed mask module is more significant for drone datasets.

### I. Results in dynamic environment

| Scene   | Total Objects | Dynamic Objects | Static Objects |
|---------|---------------|-----------------|----------------|
| Scene20 | 127           | 77              | 50             |
| Scene18 | 21            | 21              | 0              |
| Scene06 | 15            | 14              | 1              |
| Scene01 | 93            | 9               | 84             |
| Scene02 | 17            | 9               | 8              |

TABLE XI: Statistics of dynamic objects of all five scenes in VKITTI2 dataset.

Dynamic objects are a well-known challenge for visual odometry, as they violate the static-scene assumption underlying most geometric optimization methods. Although the proposed HDVO method does not explicitly model dynamics, it is still able to maintain stable localization performance in such environments. This robustness of HDVO mainly relies on the mask strategy and the direct photometric formulation, which relies on global image-level consistency rather than sparse feature correspondences, thereby reducing the influence of locally moving objects.

To analyze the impact of dynamic objects, we first analyze their distribution in the VKITTI2 dataset. Tab. XI reports the statistics of dynamic and static objects across different scenes.

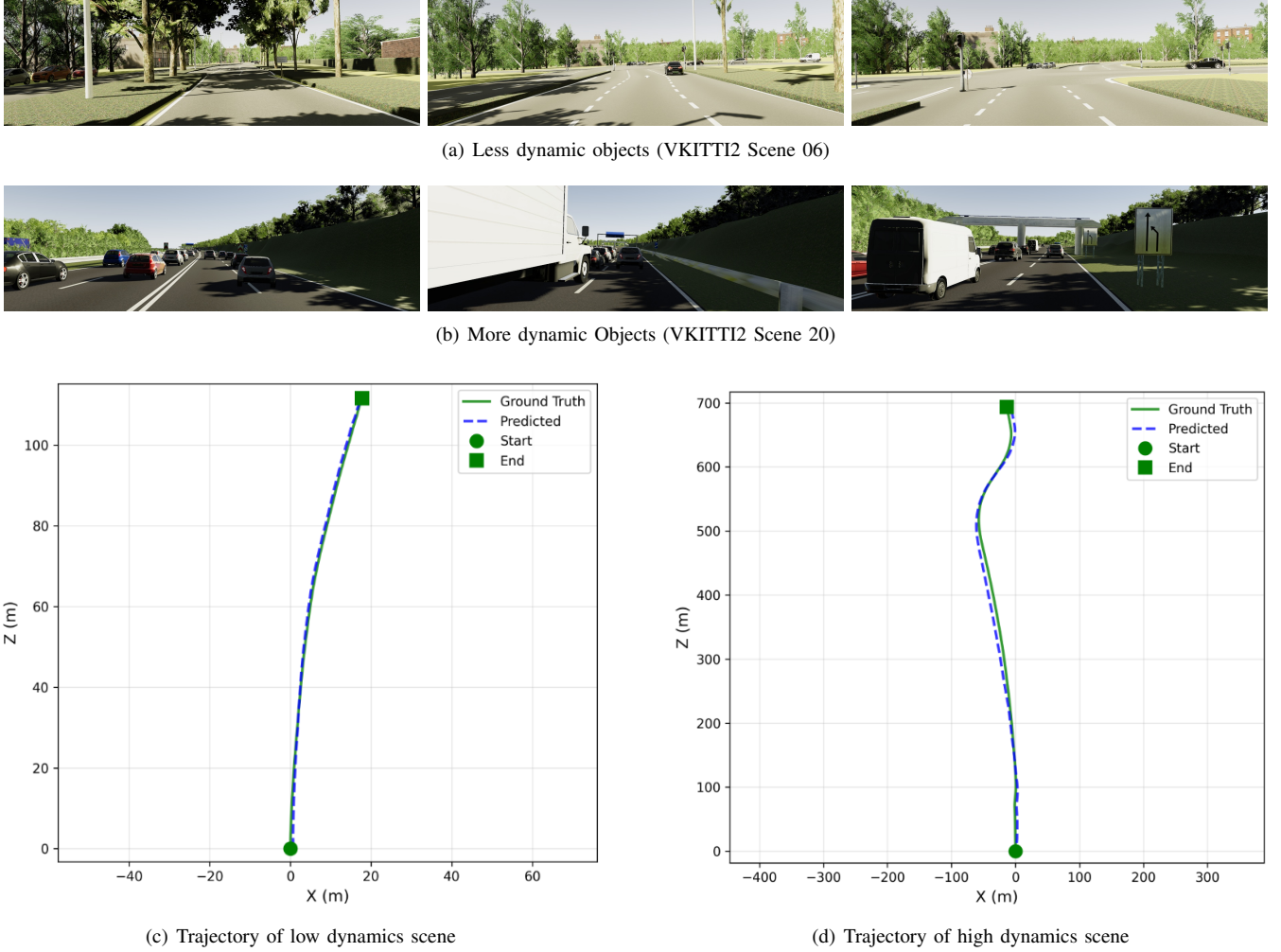


Fig. 13: Qualitative comparison of localization results under different dynamic levels. The first row shows RGB images, the second row shows the predicted trajectories compared with ground truth.

As shown, Scene 20 contains a significantly higher number of dynamic objects compared to the other scenes, whereas Scene 02 exhibits relatively low dynamic activity. Based on this observation, we select Scene 02 and Scene 20 to conduct zero-shot visual odometry experiments.

In this experiment, the HDVO model is pretrained on the KITTI Odometry dataset and directly evaluated on VKITT12 without any fine-tuning. The qualitative results are shown in Fig. 13. Although there are numerous moving vehicles in Scene 20, HDVO is still able to produce accurate and stable trajectories that closely follow the ground truth. These results indicate that the proposed method is robust to dynamic objects, even without explicit motion segmentation or dynamic scene modeling.

#### J. Results under different weather conditions

To evaluate the generalization and robustness of the proposed HDVO method, we conduct experiments on the same scene under varying weather and illumination conditions. VKITT12 is a synthetic dataset that provides six distinct weather settings for each scene, including base, overcast, rain,

| Weather  | RPE Trans (m) ↓ | RPE Rot (deg) ↓ |
|----------|-----------------|-----------------|
| Base     | 0.170           | 0.051           |
| Fog      | 0.204           | 0.066           |
| Morning  | 0.177           | 0.046           |
| Overcast | 0.183           | 0.044           |
| Rain     | 0.191           | 0.048           |
| Sunset   | 0.173           | 0.041           |

TABLE XII: Relative Pose Error (RPE) on VKITT12 Scene20 under different weather conditions.

fog, morning, and sunset. This controlled setup enables a fair and systematic comparison of visual odometry performance under diverse environmental conditions.

As illustrated in Fig. 14, we report the results on Scene 20 of the VKITT12 dataset. Scene 20 covers the longest trajectory in this dataset (711 meters over 837 frames), making it particularly suitable for evaluating long-term visual odometry stability and drift behavior.

All experiments are conducted in a zero-shot setting, where the HDVO model is pretrained on the KITTI Odometry dataset and directly evaluated on VKITT12 without any fine-tuning. The quantitative results are summarized in Tab. XII, with

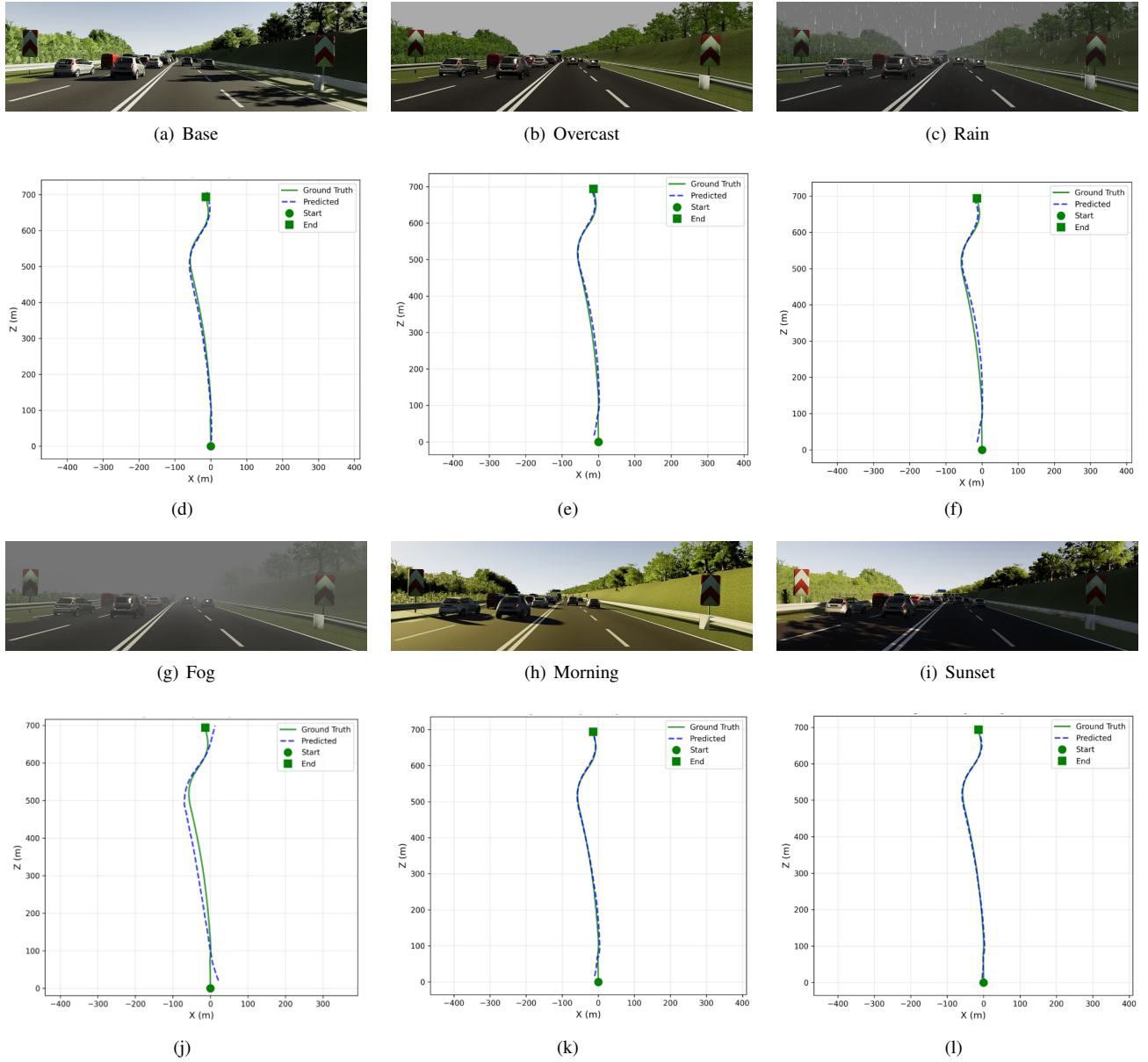


Fig. 14: Qualitative comparison of visual odometry trajectory results under six weather conditions on scene 20 of VKITTI2 dataset.

representative trajectories visualized in Fig. 14. Overall, the proposed HDVO method demonstrates consistent performance across different weather conditions. Among all settings, the fog condition exhibits the largest performance degradation, reflected by higher translational and rotational errors. This degradation can be attributed to reduced visibility and weakened geometric cues caused by dense fog. In contrast, the rain condition introduces only a slight increase in translational error, while maintaining comparable rotational accuracy.

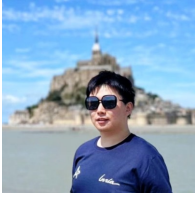
Despite these variations, the performance of HDVO under all weather conditions remains close to that of the base scene. The limited variation in relative pose error (RPE) indicates that the proposed HDVO method generalizes well and exhibits robustness to environmental changes.

#### IV. CONCLUSION

The hybrid visual odometry method proposed in this paper, named HDVO, combines model-based approaches and data-based learning in order to provide a more robust and accurate pose estimation. The main contributions of this paper lie in the brightness robust loss function and binary masks that filter the extreme noisy pixels, such as occlusion and homogeneous texture pixels. It is important to notice that data-based approaches alone provide hallucinated depth estimations that can reduce the precision of pose estimation. Therefore, a promising research direction involves introducing prior knowledge from large foundation model to identify the pixels that are not reliable or possible. Future exploration will focus on using semantic information and better depth information to enhance the quality and robustness of visual odometry.

## REFERENCES

- [1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [2] R. Mur-Artal and J. D. Tardos, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [3] A. I. Comport, E. Malis, and P. Rives, “Real-time quadrifocal visual odometry,” *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 245–266, 2010.
- [4] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2017.
- [5] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *ECCV*, 2014, pp. 834–849.
- [6] J. Engel, J. Sturm, and D. Cremers, “Semi-dense visual odometry for a monocular camera,” in *ICCV*, 2013, pp. 1449–1456.
- [7] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, “D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry,” in *CVPR*, 2020, pp. 1281–1292.
- [8] N. Yang, R. Wang, J. Stückler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry,” in *ECCV*, 2018, pp. 817–833.
- [9] 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*, 2019, pp. 8071–8081.
- [10] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, “Visual odometry revisited: What should be learnt?” in *ICRA*, 2020, pp. 4203–4210.
- [11] Z. Liu, E. Malis, and P. Martinet, “A new dense hybrid stereo visual odometry approach,” in *IROS*, 2022, pp. 6998–7003.
- [12] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *ICCV*, 2019, pp. 3828–3838. [Online]. Available: <https://github.com/nianticlabs/monodepth2>
- [13] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in *CVPR*, 2017, pp. 1851–1858. [Online]. Available: <https://github.com/tinghuiz/SfMLearner>
- [14] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *CVPR*, 2017, pp. 270–279.
- [15] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, “Learning depth from monocular videos using direct methods,” in *CVPR*, 2018, pp. 2022–2030.
- [16] 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.
- [17] R. Mahjourian, M. Wicke, and A. Angelova, “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints,” in *CVPR*, 2018, pp. 5667–5675.
- [18] E. Malis, “Improving vision-based control using efficient second-order minimization techniques,” in *ICRA*, vol. 2, 2004, pp. 1843–1848.
- [19] G. Wang, H. Wang, Y. Liu, and W. Chen, “Unsupervised learning of monocular depth and ego-motion using multiple masks,” in *ICRA*, 2019, pp. 4724–4730.
- [20] Y. Zou, Z. Luo, and J.-B. Huang, “Df-net: Unsupervised joint learning of depth and flow using cross-task consistency,” in *ECCV*, 2018, pp. 36–53.
- [21] M. Klodt and A. Vedaldi, “Supervising the new with the old: learning sfm from sfm,” in *ECCV*, 2018, pp. 698–713.
- [22] 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,” in *NeurIPS*, vol. 32, 2019.
- [23] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, “Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation,” in *CVPR*, 2019, pp. 12 240–12 249.
- [24] E. Malis and E. Marchand, “Experiments with robust estimation techniques in real-time robot vision,” in *IROS*, 2006, pp. 223–228.
- [25] Z. Liu, E. Malis, and P. Martinet, “Multi-masks generation for increasing robustness of dense direct methods,” in *ITSC*, September 2023.
- [26] J.-R. Chang and Y.-S. Chen, “Pyramid stereo matching network,” in *CVPR*, 2018, pp. 5410–5418.
- [27] 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 *CVPR*, Seattle, WA, USA, 2020, pp. 2495–2504.
- [28] G. Xu, X. Wang, X. Ding, and X. Yang, “Iterative geometry encoding volume for stereo matching,” in *CVPR*, 2023.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.
- [30] H. Hirschmüller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in *CVPR*, vol. 2, 2005, pp. 807–814.
- [31] 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*, 2017, pp. 66–75.
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [33] Openrox open-source library. [Online]. Available: <https://gitlab.inria.fr/acentaury-public/openrox>
- [34] J. Engel, J. Stückler, and D. Cremers, “Large-scale direct slam with stereo cameras,” in *IROS*, 2015, pp. 1935–1942.
- [35] Z. Yin and J. Shi, “Geonet: Unsupervised learning of dense depth, optical flow and camera pose,” in *CVPR*, 2018, pp. 1983–1992.
- [36] S. Wang, R. Clark, H. Wen, and N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in *ICRA*, 2017, pp. 2043–2050.
- [37] R. Wang, S. M. Pizer, and J.-M. Frahm, “Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth,” in *CVPR*, 2019, pp. 5555–5564.
- [38] Y. Li, Y. Ushiku, and T. Harada, “Pose graph optimization for unsupervised monocular visual odometry,” in *ICRA*, 2019, pp. 5439–5445.
- [39] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, “Towards better generalization: Joint depth-pose learning without posenet,” in *CVPR*, 2020.
- [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.
- [41] W. Wang, Y. Hu, and S. Scherer, “Tartarvo: A generalizable learning-based vo,” in *CoRL*. PMLR, 2021, pp. 1761–1772.
- [42] W. Xuan, R. Ren, S. Wu, and C. Chen, “Maskvo: Self-supervised visual odometry with a learnable dynamic mask,” in *IEEE/SICE International Symposium on System Integration (SII)*, 2022, pp. 225–231.
- [43] R. Zhu, M. Yang, W. Liu, R. Song, B. Yan, and Z. Xiao, “Deepavo: Efficient pose refining with feature distilling for deep visual odometry,” *Neurocomputing*, vol. 467, pp. 22–35, 2022.
- [44] X. Zhang, R. Zhu, R. Song, D. He, T. Yang, Z. Xiao, and B. Yan, “Losevo: Local sequence constraints for deep visual odometry,” in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2025, pp. 1–5.
- [45] J. Xie, O. Dhaouadi, W. Chen, J. Meier, J. Kaiser, and D. Cremers, “Coprou-vo: Combining projected uncertainty for end-to-end unsupervised monocular visual odometry,” in *DAGM German Conference on Pattern Recognition*. Springer, 2025, pp. 502–517.
- [46] C. Zhao, Y. Tang, Q. Sun, and A. V. Vasilakos, “Deep direct visual odometry,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7733–7742, 2021.
- [47] M. B. Azhari and D. H. Shim, “Dino-vo: A feature-based visual odometry leveraging a visual foundation model,” *IEEE Robotics and Automation Letters*, 2025.
- [48] L. Lipson, Z. Teed, and J. Deng, “Deep patch visual slam,” in *ECCV*. Springer, 2024, pp. 424–440.
- [49] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *NeurIPS*, vol. 27, 2014, pp. 2366–2374.
- [50] R. Li, S. Wang, Z. Long, and D. Gu, “Undeepvo: Monocular visual odometry through unsupervised deep learning,” in *ICRA*, 2018, pp. 7286–7291.
- [51] 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 *IROS*, Prague, Czech Republic, 2021, pp. 3542–3548.
- [52] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in *NeurIPS*, vol. 30, 2017.



**Ziming Liu** Ziming Liu is with Bosch Research, Shanghai, currently. He has obtained the computer science Ph.D. degree from INRIA institute at Sophia Antipolis, France in 2024. Before that, he has graduated from Beijing Institute of Technology (BIT) at Beijing, China in 2020, with a master degree of software engineering. His research interests lie in the intersection of autonomous driving and foundational deep learning, such as learning-based planning, localization algorithms, visual-based mapping, depth estimation, video understanding, and optimizers.



**Ezio Malis** Ezio Malis graduated both from SU-PELEC and Politecnico di Milano in 1995. He received the Ph.D. degree in Robotics and Automation from the University of Rennes, France, in 1998. After spending two years (1999–2000) as a Research Associate at the University of Cambridge (UK), he joined INRIA as a Research Scientist in Sophia Antipolis. He received in 2008 the Habilitation à Diriger des Recherches in Computer Science from the University of Nice-Sophia Antipolis. After 9 years in the ICARE and AROBAS teams, he created

the team EVOLUTION at Inria Sophia Antipolis in 2009. From 2009 til 2010 he led the EVOLUTION team to achieve technological maturity and to transfer his research work to the industry. In 2010 he co-founded the Robocortex S.A.S. company a software editor for Autonomous Robotics and of the software AugmentedPro for Augmented Reality and IoT in industrial assembling and manufacturing tasks. From 2010 to 2020 he was the CEO and the director of Research and Development at Robocortex. Since 2021, he is the leader of the ACENTAURI team at Inria. His main research interests are efficient optimization algorithms (numerical analysis, quantum computing), artificial intelligence (computer vision and machine learning) and autonomous robotics (modeling, perception, decision and control) with a peculiar emphasis on sensor-based control techniques. He is author and co-author of around a hundred papers.



**Philippe Martinet** Philippe Martinet graduated from the CUST, Clermont- Ferrand, France, in 1985 and received the Ph.D. degree in electronics science from the Blaise Pascal University, Clermont-Ferrand, France, in 1987. From 1990 til 2000, he was assistant Professor with CUST in the Electrical Engineering Department, Clermont-Ferrand. From 2000 until 2011, he has been a Professor with Institut Français de Mécanique Avancée (IFMA), Clermont-Ferrand. He was performing research at the Robotics and Vision Group of LASMEA-CNRS, Clermont-

Ferrand. In 2006, he spent one year as a visiting professor in ISRC at the Sungkyunkwan university in Suwon, South Korea. He was the leader of the group GRAVIR (over 74 person) from 2001 til 2006. From 1997 until 2011, he led the Robotic and Autonomous Complex System team (over 20 persons). From 2008 until 2011, he co-lead a Joint Unit of Technology called “Robotization in Meatee Indutry”, and the Korea France Joint Research Center on Cognitive Personal Transport Service Robot in Suwon (South Korea). In September 2011, he moves to Ecole Centrale de Nantes and make his research at IRCCYN in the robotics team. Until 2015, he stayed associated researcher to Pascal Institute in Clermont-Ferrand. Since 2017, IRCCyN becomes LS2N. He was deputy head of the ARMEN Team. In November 2017, he moves to Inria Sophia Antipolis as Research director. From 2018 until 2021, he was the leader of the CHORALE team. In mai 2021, he becomes member of ACENTAURI team. He was coordinator of the EMJMD EMARO+ from 2014 until 2016. Until 2017, he was in charge of the robotics speciality in the ARIA master. From 2015 until 2019, he was deputy head of the French robotics research Network. From 1990, he has driven four habilitations, 27 (+ 13 as co-supervisor) past Ph.d students, plus 3 (+ 1 as co-supervisor) in progress, 33 Master students and 5 Post-doc. For the same period, he is author and co-authors of more than three hundred sixty references 61 IJ, 2 NJ, 216 IC, 27 NC, 52 IW, 1 NW, and 8 book Chapters.