SELF-SUPERVISED SVDE FROM VIDEOS WITH DEPTH VARIANCE TO SHIFTED POSITIONAL INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, much attention has been drawn to learning the underlying 3D structures of a scene from monocular videos in a fully self-supervised fashion. One of the most challenging aspects of this task is handling the independently moving objects as they break the rigid-scene assumption. For the first time, we show that pixel positional information can be exploited to learn SVDE (Single View Depth Estimation) from videos. Our proposed moving object (MO) masks, which are induced by depth variance to shifted *positional information* (SPI) and referred to as 'SPIMO' masks, are very robust and consistently remove the independently moving objects in the scenes, allowing for better learning of SVDE from videos. Additionally, we introduce a new adaptive quantization scheme that assigns the best per-pixel quantization curve for our depth discretization. Finally, we employ existing boosting techniques in a new way to further self-supervise the depth of the moving objects. With these features, our pipeline is robust against moving objects and generalizes well to high-resolution images, even when trained with small patches, yielding state-of-the-art (SOTA) results with 4 to 8$\times$ fewer parameters than the previous SOTA that learns from videos. We present extensive experiments on KITTI and CityScapes that show the effectiveness of our method.

1 INTRODUCTION

Estimating depth from a single image has multiple applications ranging from robotics, navigation, and computational imaging. However, learning single view depth estimation (SVDE) from monocular videos without ground truth depths is challenging as camera poses and depths are estimated up to an unknown and inconsistent scale. In addition, by the nature of self-supervision from reprojection error, objects moving at the same speed as the camera will have no disparities thus will be assigned infinite depths. On the other hand, objects moving faster than the camera will have larger disparities, resulting in implausibly closer depths. In general, dynamically moving objects are likely to cause incorrect depths along their boundaries due to their weak or in-existent 3D self-supervision.

Traditionally, deep-learning-based pipelines (Zhou et al., 2017; Godard et al., 2019; Guizilini et al., 2020a) for learning SVDE from videos rely on: (i) a CNN that directly estimates inverse depth; (ii) a CNN that directly regresses relative camera poses; and (iii) a backward warping-based loss function that measures the similarity between the target center frame and the warped reference frames. Independently moving objects in such loss functions are commonly removed by moving object masks that are computed by analyzing the reconstruction errors between the target frame and the multiple warped reference frames (Godard et al., 2019). In contrast, we present a novel learning pipeline that relies on pixel positional information to compute the moving object masks and forward warping-based loss functions (image synthesis) for learning accurate SVDE from high-resolution monocular videos. We summarize our contributions as:

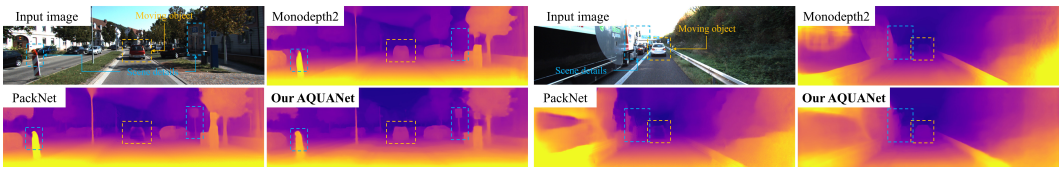

Figure 1: Our method consistently learns more details and holes-free dense depths from videos.

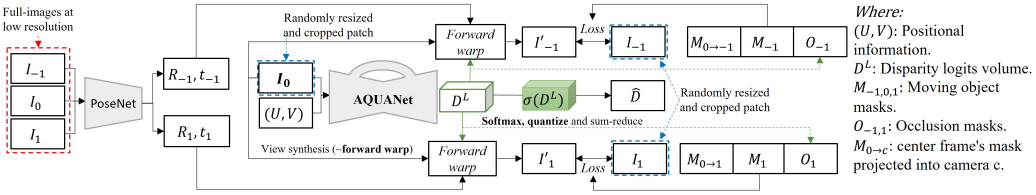

Figure 2: Proposed pipeline for learning SVDE from videos.

- **Robust estimation of moving object masks by shifted positional information.** To ignore independently moving objects (that can break the rigid-scene assumption) we propose novel moving object masks induced by depth variance to shifted *positional information*, referred to as SPIMO masks. These masks are more robust and less sparse than the traditional auto-masks (Godard et al., 2019), allowing for better learning of SVDE from mono-videos. In addition, SPIMO masks do not require any additional data such as semantic masks, optical flow, nor domain specific assumptions.
- **Adaptive disparity quantization.** Disparity / depth discretization converts the harder regression task into an easier classification task. However, in contrast with previous works that use linear or exponential quantization (Xie et al., 2016; Gonzalez & Kim, 2021) for stereoscopic view synthesis, we propose a novel adaptive quantization scheme for free view synthesis. In our quantization scheme, the proposed Adaptive QUAntization network, AQUANet, automatically learns to assign a quantization curve that dedicates more distant sampling levels if the target pixel is likely to be far, and a quantization curve with more close-by sampling levels if it is closer to the camera, improving the accuracy of the final aggregated depth map.

2 Related Work

2.1 Fully Supervised Learning of SVDE.

The early fully-supervised methods for SVDE utilized deep neural networks to perform depth map regression. For example, in (Eigen et al., 2014), one network estimated global depth predictions, and the other network refined the predictions with local context. To address the slow convergence of regression models, discretizing depth values allowed the SVDE task to be an ordinal regression problem (Fu et al., 2018). To further improve the quality of high-frequency details in SVDE, a double-estimation method for merging depth estimations at different resolutions was proposed by Miangoleh et al. (2021). The resulting depth maps contained high-frequency details of the image while preserving the scene's structural consistency. While the fully-supervised learning of the SVDE task has been gaining lots of attention, both the tasks of depth estimation and completion have been explored by Guizilini et al. (2021). Although many methods are proposed for fully supervised learning of SVDE, the burden of acquiring the depth ground truths limits the scalability of the training dataset and the generalization of models.

2.2 Self-Supervised Learning from Stereo.

Without the depth ground truths, a neural network can learn to estimate depth by minimizing a photometric loss between the right (or left) views and a synthetic view when the training data consists of synchronized stereo images. To further supervise a CNN for SVDE from stereo, semi-global matching (SGM (Hirschmuller, 2005)) pseudo-GTs have also been explored by Watson et al. (2019); Tosi et al. (2019). In the recent work of Zhu et al. (2020), segmentation masks have also been used in contour consistency losses to generate depth maps with more accurate borders.

The recent work of GonzalezBello & Kim (2020) proposed a multi-view occlusion module to account for the occluded regions on the reconstruction losses and an exponential disparity discretization which led to considerable performance improvements. In (Gonzalez & Kim, 2021), the authors introduced 'neural positional encoding' (NPE) and a matting laplacian loss. NPE projects per-pixel positional information into a higher dimensional space via a fully-connected network to let the CNN reason about location-specific image characteristics (e.g., projection distortions, ground-vs-sky, etc.). Their matting laplacian loss guides the CNN to produce sharper depth boundaries by exploiting the sharpening properties of natural image matting (Levin et al., 2007).

2.3 SELF-SUPERVISED LEARNING FROM VIDEOS.

Learning self-supervised SVDE from videos is a much more challenging task. While fixed stereo images are synchronized, video sequences have temporal distances and arbitrary camera poses between frames. This hinders the estimation of moving object depths and demands a secondary network for camera pose estimation as proposed by Zhou et al. (2017). In their work, 'explainability' masks were *estimated* to handle such moving objects.

In Monodepth2 (Godard et al., 2019), the authors proposed auto-masking, which selects the pixels with the smallest projection errors to remove moving objects in the scene, improving SVDE performance. However, there is still a limitation on moving objects which move faster or slower than the camera. The recent work of Guizilini et al. (2020b) explored filtering out the sequences that have dynamic objects using a two-stage training strategy, semantics, and the PackNet Guizilini et al. (2020a) backbone. However, sub-sampling the training sequences can reduce the model's generalization capabilities. Similarly, the work of Klingner et al. (2020) utilized a semantic masking scheme by cross-domain training of semantic segmentation and SVDE. Moreover, several methods (Wang et al., 2021b; Jung et al., 2021b; Wang et al., 2021a; Liu et al., 2021; Shu et al., 2020; Lyu et al., 2020) have improved the depth estimation quality, but there is still a lot of room for improvement. This paper suggests the SPIMO masks, which keep the model's scalability without requiring any additional data from different tasks. Using the SPIMO masks, we can successfully exclude moving objects for loss calculation and achieve outperforming results.

3 PROPOSED METHOD

Our training strategy is **two-staged**. Firstly, we let the SVDE network learn from all pixels (static and dynamic pixels) in the target and reference frames with an occlusion-aware synthesis loss following (GonzalezBello & Kim, 2020). Secondly, we devise a new method for detecting dynamically moving objects by measuring the dispersion of multiple depth estimates from the first stage network under several *positional information perturbations* (see Section 3.3). In the second training stage, such robust moving object masks are used to effectively train an SVDE network from scratch, by ignoring the independently moving pixels in the scene in our new occlusion and moving-object-aware synthesis loss (see Section 3.5). Additionally, we exploit recent advances in SVDE (Miangoleh et al., 2021; Gonzalez Bello & Kim, 2021) to self-supervise the moving object regions with "boosted" depth estimates, further improving our results (see Section 3.4).

3.1 NETWORK ARCHITECTURE

Our new proposed pipeline is depicted in Fig. 2. The target frame (I_0) is projected via free-view synthesis (or forward warping) onto the reference frames (I_{-1}, I_1), generating new synthetic views seen from the reference camera positions. Forward warping is achieved by following the FAL-Net (GonzalezBello & Kim, 2020), which can be generalized to any camera position and augmented for *adaptive* inverse depth discretization (see Section 3.2 for more details). The synthetic views, denoted by I'_{-1} and I'_1 in Fig. 2, are compared against the GT reference views with geometrically inspired reconstruction losses. Note that these losses not only consider masking the occluded contents in the target frame that are visible in the reference views but also the moving objects in the reference frames and the projected target frame, as denoted by the three masks in each reconstruction loss (e. g. M_1, $M_{0\to1}$, and O_1 for frame I_{-1}).

As depicted in Fig. 2, learning SVDE requires at least two branches: one for disparity and the other for relative pose estimation. The first is carried out by our Adaptive QUAntization Network, referred to as 'AQUANet', which takes as input a single view $\mathbf{I}_0$ and its corresponding pixel positional data $(\mathbf{U}, \mathbf{V})$ and produces the disparity logit volume $\mathbf{D}^L$ and the per-pixel adaptive quantization parameter β. AQUANet adopts the network backbone in (Gonzalez & Kim, 2021), which is a simple auto-encoder with skip connections that incorporates neural positional encoding (NPE). NPE maps pixel coordinates (u, v) to a higher dimensional space via a fully connected network to be further processed by the CNN's encoder. On the other hand, PoseNet (Zhou et al., 2017) is a convolutional encoder that takes target and source images as input and maps them into relative camera extrinsics (x, y, and z rotations and translations).

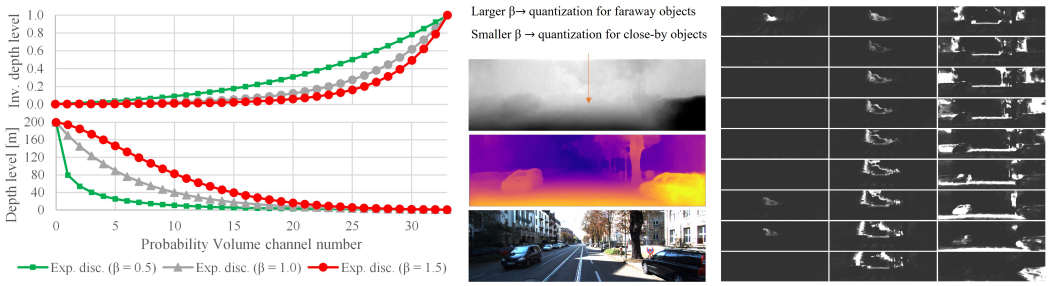

Figure 3: Adaptive quantization. (Left) Quantization curves with adaptive parameter β. (Center) From bottom to top: Input image, disparity, and β map. (Right) Disparity probability volume.

3.2 Adaptive Disparity Quantization for View Synthesis

An approximation to forward warping an input image $\mathbf{I}_0$ into a new camera position c can be achieved by progressively warping $\mathbf{I}_0$ into c by N depth planes and summing the projected images weighted by their per-pixel depth probability distribution. We achieve this by generalizing and improving the synthesis method in (GonzalezBello & Kim, 2020), which is originally devised for fixed stereoscopic view synthesis, by incorporating camera intrinsics ($\mathbf{K}_0$ and $\mathbf{K}_c$) and extrinsics (R_c and t_c) into the warping operation $g(\cdot)$. The synthetic view seen from camera position c can then be expressed by

$$\mathbf{I}'_c = \sum_{n=0}^{N-1} g\left(\mathbf{I}_0, d_n, R_c, t_c, \mathbf{K}_0, \mathbf{K}_c\right) \odot \mathbf{D}_n^{P_{0\to c}}, \tag{1}$$

where d_n is the inverse depth quantization level and $\odot$ denotes the Hadamard product. $\mathbf{D}^{P_{0\to c}}$ is the disparity probability volume projected into camera c, described by

$$\mathbf{D}^{P_{0\to c}} = \sigma\left(\left[g\left(\mathbf{D}_n^L, d_n, R_c, t_c, \mathbf{K}_0, \mathbf{K}_c\right)\right]_{n=0}^{N-1}\right), \tag{2}$$

where σ is the channel-wise softmax operation and $[\cdot]$ is the channel-wise concatenation operation. $\mathbf{D}^L$ is the disparity logit volume which can be soft-maxed, quantized, and sum-reduced to obtain the final disparity estimate

$$\hat{\mathbf{D}} = \sum_{n=0}^{N-1} d_n \sigma(\mathbf{D}^L)_n. \tag{3}$$

In (GonzalezBello & Kim, 2020), the authors defined $d_n = d_{max} e^{(n/N-1)\ln d_{max}/d_{min}}$ as a fixed exponential quantization of disparity. Such quantization scheme distributes the sampling levels more uniformly in the depth domain as described by the gray line in Fig. 3. However, this assumes that all pixels in the input image will follow the same distribution. Properly allocating 'depth bins' to each pixel can distribute probabilities more intensively in the region of interest, generating a much more accurate disparity map. For example, a more "linear" inverse depth quantization could benefit the depth estimation of close-by objects by assigning more sampling levels to larger disparities as shown by the green line in Fig. 3. On the other hand, a steeper inverse depth quantization should improve the depth estimation of distant objects, as more bins are assigned to smaller disparities, as depicted by the red line in Fig. 3. The quantization curves depicted in Fig. 3 belong, in fact, to the same family of curves controlled by the adaptive quantization parameter β, which we propose to incorporate in a per-pixel ($\boldsymbol{p} = (u, v)$) manner in our new adaptive quantization scheme, given by

$$\boldsymbol{d}(\boldsymbol{p}) = \left[d_{max} e^{\ln(d_{max}/d_{min})\left((n/N)^{\beta(\boldsymbol{p})}-1\right)}\right]_{n=0}^{N}. \tag{4}$$

Then, our final inverse depth estimate is described by $\hat{\mathbf{D}} = \boldsymbol{d} \cdot \sigma(\mathbf{D}^L)$ (5), where $\cdot$ denotes the channel-wise dot product. Note that in (GonzalezBello & Kim, 2020), d_n is a scalar value, while $\boldsymbol{d}$ is a tensor in our proposed adaptive quantization. It is also worth noting that the hypothetical inverse depth planes in Eqs. 1 and 2, that were originally single scalar values in (GonzalezBello & Kim, 2020), become hypothetical inverse depth maps with different values per-pixel location by our adaptive quantization scheme. Fig. 3 provides a visualization of the learned adaptive quantization. The adaptive parameter β controls the per-pixel disparity quantization curve as depicted in the center of Fig. 3. As can be observed, β effectively assigns the most suitable quantization curve for far and close-by object pixels. The unquantized $\sigma(\mathbf{D}^L)$, depicted to the right of Fig. 3, effectively discretizes the estimated disparity in several levels.

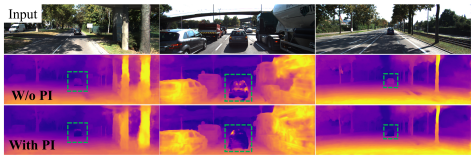

Figure 4: Moving object artifacts w/o PI.

Figure 5: Perturbations for Moving object masks.

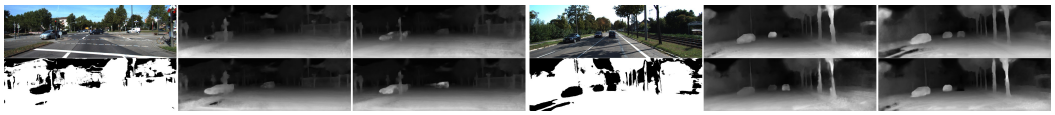

Figure 6: Single image inputs, disparity estimates from multiple forward passes (with $N = 4$ different positional information offsets), and computed SPIMO masks.

The previous work (Bhat et al., 2021) also explored adaptive discretization of depth for the fully supervised case of SVDE. However, while they generate N independently adaptive depth bins globally shared across all pixel locations, we propose a per-pixel quantization so that each output pixel has its own quantization curve. One weakness of AdaBins (Bhat et al., 2021) is that it requires GT depths to supervise the bin centers, making it not suitable for the task of interest in the present manuscript, self-supervised learning of SVDE. Furthermore, note that our adaptive quantization scheme not only serves as a disparity representation, but we can also efficiently perform forward warping with it.

3.3 Computation of Moving Object Masks with Shifted Positional Information

The neural positional encoding (NPE) proposed in the PLADE-Net Gonzalez & Kim (2021) was known to be simple yet effective in learning SVDE from stereo images as it provides the network with the means of understanding the relative location of a random crop with respect to the full image. However, when learning from monocular videos, the NPE facilitates the network to memorize the potential locations and appearances of independently moving objects (e.g., cars) in the scene. This is not desirable for the generalization of SVDE as it can overly accentuate the effects of moving objects resulting in incorrect depth estimates. These adverse effects are shown in Fig. 4 and Table 1, where the AQUANet with positional information (PI) yields strong artifacts on the moving objects indicated by the green boxes and slightly worse metrics than the AQUANet without PI.

However, by exploiting this behavior, we can *fool* a fully trained network to recover the geometries of dynamically moving objects. We *fool* the network by feeding the shifted positional information into the NPE, which enforces the network to switch from an *overfitting* mode into a *generalization* mode, by producing different depth estimates on the regions that were exposed to a weaker 3D self-supervision (moving objects, highly homogeneous regions, etc.).

In this paper, for the first time, we propose a novel way to compute moving object masks by exploiting the recovered geometries obtained with the shifted positional information. We propose to measure the depth variance per-pixel to classify it as a likely moving or static object pixel in our Shifted Positional Information Moving Object (SPIMO) masks. Interestingly, our proposed approach allows for obtaining a *likely* moving object mask from a single image rather than a moving object mask from multiple images. Based on this approach, neither an extra network for estimating explainability masks (Zhou et al., 2017) nor a dedicated network for computing optical flows is needed. Instead, the shifted positional information is alone enough to tackle the ill-posed problem of learning SVDE from videos.

To obtain our SPIMO masks we first build a perturbed depth volume $\mathbf{D}^v$, with multiple network passes under shifted positional information, as shown in Fig. 6 and described by:

$$\mathbf{D}^v = \left[\text{AQUANet}_F(\mathbf{I}, (\mathbf{U} + u_i, \mathbf{V} + v_i))^{-1}\right]_{i=1}^{N}, \tag{6}$$

where each of its N channels is a depth estimate with positional offsets (u_i, v_i) and AQUANet_F denotes a fixed copy of our network after the first training stage. Given normalized positional coordinates (image center at $(0, 0)$, top-left corner at $(-1, -1)$), we empirically set $u = [0, 0.5, -0.5, 0]$ and $v = [0, 0, 0, -0.25]$ positional offsets. Note that we use depth instead of inverse depth in $\mathbf{D}^v$ as the variations between close and far away estimates are more evident in depth units. The SPIMO

Figure 7: Visual comparison between our shifted positional information moving object (SPIMO) masks and those computed following Monodepth2 (Godard et al., 2019).

mask is then given by applying a threshold γ on the normalized channel-wise variance as given by

$$\mathbf{M} = \left\{1 \; if \; \textstyle\sum_{i=1}^{N}(\mathbf{D}_i^v - \bar{\mathbf{D}}^v)^2/((\bar{\mathbf{D}}^v)^2(N-1)) < \gamma, \; 0 \; o.w.\right\}, \tag{7}$$

which is similar to the index of dispersion. We empirically set $\gamma = 3\%$ for all our experiments as it was observed to drop most moving object pixels while maintaining rigid object pixels. We also explored other low-level perturbations and found them to be not effective in generating depth variances correlated to moving objects (MOs). To demonstrate it, we added random noise and RGB channel swap perturbations, but failed in masking out the MOs, as can be seen in the 2^{nd} and 3^{rd} results of Fig 5. Only perturbations on PI can induce enough depth variance on the MO depths to differentiate them, as shown in the last result of Fig.5. This is because: (i) PI shifts do not degrade the input images; and (ii) the networks learn the potential locations and appearances of MOs.

3.4 Boosting as self-supervision for moving objects

We observed that the quality of moving object depths (e.g., cars, bikers) produced by the second training stage with SPIMO masks was much lower than that of the static objects (e.g., roads, trees, traffic signs). This is due to the network not being exposed to those regions during training and such masked regions being only penalized by the smoothness loss. To prevent this, we propose to use a watered-down version of the boosting techniques in (Miangoleh et al., 2021; Gonzalez Bello & Kim, 2021) to self-supervise the moving objects in the masked regions. We define the disparity estimates from a previous-epoch copy of the AQUANet under training at full, reduced and increased resolutions as $\hat{\mathbf{D}}_F = \text{AQUANet}'(\mathbf{I}_0, (\mathbf{U}, \mathbf{V}))$, $\hat{\mathbf{D}}_{F\downarrow} = \frac{4}{3}H(\text{AQUANet}'(H(\mathbf{I}_0, \frac{3}{4}), H((\mathbf{U}, \mathbf{V}), \frac{3}{4})), \frac{4}{3})$, and $\hat{\mathbf{D}}_{F\uparrow} = \frac{4}{5}H(\text{AQUANet}'(H(\mathbf{I}_0, \frac{5}{4}), H((\mathbf{U}, \mathbf{V}), \frac{4}{5})), \frac{4}{5})$, respectively. Where $H(\cdot)$ denotes the resizing operation. Then, the boosted disparity can be computed by a selective combination of $\hat{\mathbf{D}}_F$, $\hat{\mathbf{D}}_{F\downarrow}$, and $\hat{\mathbf{D}}_{F\uparrow}$, as

$$\mathbf{D}^* = (\hat{\mathbf{D}}_F + \bar{\mathbf{D}}_F \odot \hat{\mathbf{D}}_{F\downarrow} + (1 - \bar{\mathbf{D}}_F^2) \odot \hat{\mathbf{D}}_{F\uparrow})/(1 + \bar{\mathbf{D}}_F + (1 + \bar{\mathbf{D}}_F^2)) \tag{8}$$

where $\bar{\mathbf{D}}_F$ is the normalized mean disparity.

$$\bar{\mathbf{D}}_F = (\hat{\mathbf{D}}_F + \hat{\mathbf{D}}_{F\downarrow} + \hat{\mathbf{D}}_{F\uparrow})/max(\hat{\mathbf{D}}_F + \hat{\mathbf{D}}_{F\downarrow} + \hat{\mathbf{D}}_{F\uparrow}). \tag{9}$$

Eq. 8 effectively blends disparities from the fixed network, assigning more weights on the upscaled estimates for faraway objects and more weights on the downscaled estimates for the close-by objects, producing a slightly better depth map (Miangoleh et al., 2021; Gonzalez Bello & Kim, 2021).

3.5 Loss Functions

In the first training stage, we learn from all non-occluded image pixels with a combination of occlusion-aware synthesis l_s^o and edge-aware disparity smoothness l_{ds} losses, as given by $l_{st1} = l_s^o + \alpha_{ds} l_{ds}$, where $\alpha_{ds} = 0.1$ following (Godard et al., 2017).

In the second stage, we train the network from scratch with an occlusion and moving object-aware synthesis loss l_s^{OM}, l_{ds}, and a boosting loss l_b, as given by

$$l_{st2} = l_s^{om} + \alpha_{ds} l_{ds} + \alpha_b l_b, \tag{10}$$

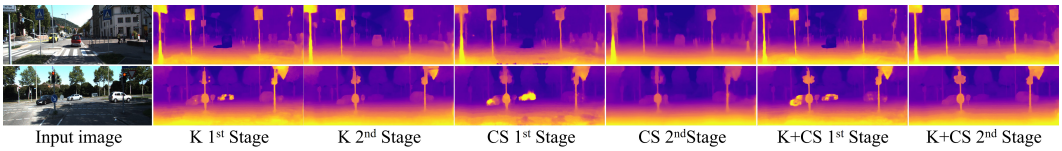

Figure 8: AQUANet results on KITTI images for 1st and 2nd training stages.

where $\alpha_b = 0.1$ and $\alpha_{ds} = 0.2$ are set empirically. Our synthesis loss, $l_s^{om} = 0.5l_{s_1}^{om} + 0.5l_{s_{-1}}^{om}$, averages the contribution from previous and next reference views.

Occlusion / moving object-aware synthesis loss. Given the combination of occlusion and SPIMO masks $\mathbf{W}_1 = \mathbf{O}_1 \odot \mathbf{M}_1 \odot \mathbf{M}_{0\to1}$, where $\mathbf{M}_{0\to1}$ is the target frame's SPIMO mask projected into the reference camera, this term is defined as

$$l_{s_1}^{om} = ||\mathbf{W}_1 \odot (\mathbf{I}'_1 - \mathbf{I}_1)||_1 + \alpha_p \sum_{l=1}^{3} ||\phi^l(\mathbf{I}'_1) - \phi^l(\mathbf{I}_1^w)||_2^2, \tag{11}$$

where $\mathbf{I}_1^w = (1-\mathbf{W}_1)\odot\mathbf{I}'_1 + \mathbf{W}_1 \odot \mathbf{I}_1$ is the GT view with moving objects replaced by those in $\mathbf{I}'_1$. The weight $\alpha_p = 0.01$ is set empirically to balance the contribution between the L1 and perceptual losses (Johnson et al., 2016) and $\phi^l(\cdot)$ denotes the l^{th} maxpool layer of a pre-trained VGG19 (Simonyan & Zisserman, 2014). Note that $\mathbf{O}_1$ is obtained from $\mathbf{D}^L$ following the method in (GonzalezBello & Kim, 2020). l_s^o in l_{st1} is obtained by just setting $\mathbf{W} = \mathbf{O}$ in l_s^{om}.

Boosting loss. This loss builds on the previous works (Miangoleh et al., 2021; Gonzalez Bello & Kim, 2021) and provides refined disparity maps $\mathbf{D}^*$ to self-supervise the moving object regions by

$$l_b = \frac{1}{max(\mathbf{D}^*)} ||(1 - \mathbf{M}_0) \odot (\hat{\mathbf{D}} - \mathbf{D}^*)||_1. \tag{12}$$

Note that the boosting loss is only applied to the pixel regions that are likely moving independently, normalized by the maximum disparity value $max(\mathbf{D}^*)$.

4 Experiments and Results

We trained our novel pipeline in two stages for 110 and 55 epochs for networks trained from scratch and networks with pre-trained backbones, respectively. For the first 5 epochs, we kick-start the PoseNet with a 'naive disparity map'. This map is a vertical gradient image where the first and last image rows are assigned zeros and ones, respectively. We implemented our pipeline with PyTorch and trained it on an NVIDIA A100 GPU with a batch size of 8. We optimized our networks using the default Adam (Kingma & Ba, 2014) optimizer with an initial learning rate of 0.0001. The learning rate is halved at [55%, 70%, 90%] of the total iterations. We applied random cropping (for all datasets) of 192×640 and random resizing from 0.5 to 1.5 (see Appendix A.2 for more details). We set the disparity discretization levels as 33 or 49, which well fit our adaptive quantization scheme.

4.1 Datasets

KITTI (K) (Geiger et al., 2012). To compare our results with a wide range of previous works, we utilized the Eigen train and test splits (Eigen et al., 2014) of the KITTI dataset. The train split contains 22,600 384×1208 image sequences captured at 10Hz. The test split contains 697 and 652 images with projected sparse LiDAR GTs (which we cap at 80*m*), respectively in its original and improved versions (Uhrig et al., 2017). The latter aggregates multi-views for higher-quality depths.

CityScapes (CS) (Cordts et al., 2016). We utilize the train folder of the CS dataset, which contains ∼3K 384×1208 images, each surrounded by 29 frames captured at 17Hz and adding up to ∼80K training sequences. To ensure large enough motion in the sequences, we skip every other frame, reducing the frame rate to half.

4.2 Ablation Studies

Table 1 shows the results of extensive ablation studies on the KITTI dataset to inspect the effectiveness of each component in our method. See Appendix A.3 for additional ablation experiments.

Experiment 1 (Expmt. 1) in Table 1. We validate our SPIMO masks by training our AQUANet for SVDE without masking any moving object, denoted as AQUANet(1st). The various KITTI metrics

Table 1: Ablation studies of our AQUANet on KITTI(Geiger et al., 2012). Metrics defined in (Eigen et al., 2014). Error metrics are the lower the better and accuracy metrics are the higher the better.

Expmt.	Methods	abs rel	sq rel	rmse	rmse_{log}	δ^1	a^2	a^3
1	AQUANet (1st, w/o PI)	0.118	1.157	4.099	0.164	0.890	0.974	0.988
	AQUANet (1st)	0.120	1.457	4.286	0.167	0.891	0.970	0.986
	AQUANet (1st-stat)	0.080	0.287	2.858	0.114	0.936	0.992	0.999
2	AQUANet (with (Godard et al., 2019) masks)	0.113	0.820	4.236	0.167	0.881	0.970	0.989
	DIRNet-BW	0.098	0.579	3.910	0.149	0.900	0.976	0.992
	vPLADE-Net	0.084	0.366	3.240	0.124	0.930	0.987	0.997
	AQUANet	0.080	0.328	3.095	0.118	0.934	0.989	0.997
3	AQUANet($a_b = 0.1$)	0.078	0.320	3.117	0.117	0.935	0.989	0.997
	AQUANet(full)	0.077	0.324	3.032	0.115	0.938	0.989	0.997
	AQUANet_{R34}(full)	0.070	0.285	2.988	0.107	0.946	0.991	0.998

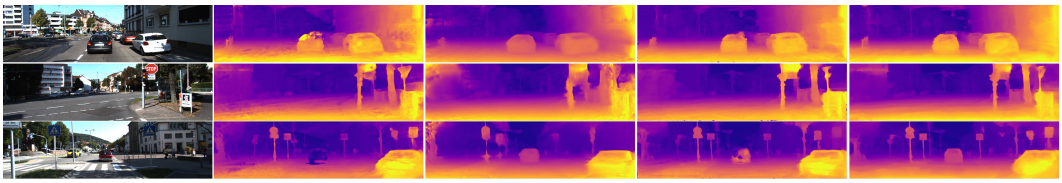

Figure 9: Qualitative comparison among different network architectures.

(Eigen et al., 2014) are poor due to the moving objects for which infinite or incorrect depths are often predicted, as depicted in the 2nd column of Fig. 8. We then measure the performance of AQUANet(1st) on the non-moving objects only, by masking moving objects with computed SPIMO masks, denoted as AQUANet(1st-stat). The better metrics of AQUANet(1st-stat) imply that our SPIMO masks can effectively remove the scene's moving objects.

Expmt. 2. We inspect our second training stage on different architectures with similar network backbones. Firstly, we train our AQUANet with 'auto-masks' from (Godard et al., 2019). As can be noted in Table 1 and Fig. 9, the auto-masks are not enough to entirely remove the moving objects in the scene, forcing the AQUANet with auto-masking to generate holes in the estimated disparity maps and considerably lower quantitative performance. Fig. 7 depicts our SPIMO masks for various KITTI (Geiger et al., 2012) sequences and their corresponding auto-masks following (Godard et al., 2019). As observed, SPIMO masks are much less sparse and remove the independently moving objects better, while the photometric assumptions in (Godard et al., 2019) do not always manage to remove the moving objects in the scene.

Secondly, the DIRNet-BW, which learns using backward warping-based loss functions (as in (Godard et al., 2019)), but with our SPIMO masks, manages to estimate full depth maps but struggles in estimating plausible depths for the thin objects in the scene. This is reflected in the improved metrics in Expmt. 2 of Table 1 and qualitative results of Fig. 9. Thirdly, the vPLADE-Net, a video extension of (Gonzalez & Kim, 2021) with our SPIMO masks yields improved metrics and more detailed depth maps but presents disparity artifacts near the image borders (such as small holes) and incorrect depths around the scene's objects boundaries due to the limitations of the fixed quantization, as shown in Table 1 and Fig. 9. Finally, our AQUANet, with SPIMO masks and pixel-wise adaptive disparity quantization, remarkably obtains the best results in Expmt. 2, both quantitatively and qualitatively, as depicted in Table 1 and Fig. 9. Finally, The considerable differences between the first and second training stages on the different datasets can be observed in Fig. 8.

Expmt. 3. We ablate the effects of boosting into our training. Our AQUANet with boosting ($a_b = 0.1$) achieves improved results in most metrics. However, we observed that even better results could be achieved if the boosted depths $\mathbf{D}^*$ are incorporated into the SPIMO mask computation process in Eq. 6, denoted as AQUANet(full). Furthermore, we ablated the effects of adopting the Resnet-34 as a bigger and pre-trained encoder backbone into our AQUANet(full), denoted as AQUANet_{R34}(full). Our AQUANet_{R34}(full) yields the SOTA result by a large margin.

4.3 Results

KITTI. A quantitative comparison between our method and existing SOTA methods that learn from videos is presented in Table 2 (See Appendix A.5 for full table). Our AQUANet always achieves the

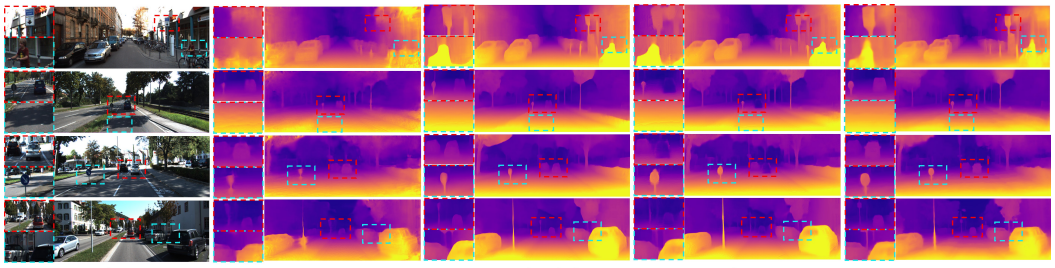

Figure 10: Qualitative comparisons on KITTI. Our AQUANet consistently estimates more detailed depths and is robust against moving objects thanks to our SPIMO masks.

Table 2: Results on the KITTI Eigen Split Eigen et al. (2014). V, V_{se}: Self-supervised from video or video+semantics. Models are trained on KITTI (K) or CityScapes (CS). Par: # of params.

	Reference	Methods	Sup	Data	Par	abs rel	sq rel	rmse	$\log_{rmse}$	a^1	a^2	a^3
Original Split	Godard et al. (2019)	Monodepth2	V	K	14	0.115	0.882	4.701	0.190	0.879	0.961	0.982
	Guizilini et al. (2020a)	PackNet	V	K	120	0.107	0.802	4.538	0.186	0.889	0.962	0.981
	Lyu et al. (2020)	HR-Depth	V	K	14	0.104	0.727	4.410	0.179	0.894	**0.966**	**0.984**
	Proposed	*AQUANet*	V	K	14	0.115	**0.656**	**4.251**	0.186	0.875	0.959	0.983
	Guizilini et al. (2020a)	PackNet	V	CS→K	120	0.104	0.758	4.386	0.182	0.895	0.964	0.982
	Guizilini et al. (2020b)	Guizilini *et al.*	V_{se}	CS→K	140	**0.100**	0.761	4.270	**0.175**	**0.902**	0.965	0.982
	Proposed	*AQUANet*	V	K+CS	14	0.112	0.669	4.263	0.182	0.882	0.963	**0.984**
Improved Split	Proposed	*AQUANet*(1^{st})	V	CS	14	0.179	1.881	5.396	0.240	0.790	0.933	0.969
	Proposed	*AQUANet*	V	CS	14	0.131	0.786	4.686	0.186	0.837	0.962	0.988
	Godard et al. (2019)	Monodepth2	V	K	14	0.092	0.536	3.749	0.135	0.916	0.984	0.995
	Guizilini et al. (2020a)	PackNet	V	K	120	0.078	0.420	3.485	0.121	0.931	0.986	0.996
	Proposed	*AQUANet*	V	K	14	0.079	0.324	3.032	0.115	0.938	0.989	0.997
	Proposed	*AQUANet*$_{R34}$	V	K	25	**0.070**	0.285	2.988	**0.107**	0.946	0.991	**0.998**
	Guizilini et al. (2020a)	PackNet	V	CS→K	120	0.071	0.359	3.153	0.109	0.944	0.990	0.997
	Proposed	*AQUANet*	V	K+CS	14	0.076	0.301	**2.921**	0.111	0.943	0.991	**0.998**
	Proposed	*AQUANet*$_{R34}$	V	K+CS	25	**0.070**	**0.283**	2.943	**0.107**	**0.948**	**0.992**	**0.998**

best sq rel and RMSE metrics in comparison with the previous self-supervised methods when trained with (K) only and with (K+CS) for both the original and the improved Eigen test splits (Eigen et al., 2014). For the improved Eigen test split (which contains much denser depths and is therefore more reliable), our AQUANet outperforms the SOTA method of PackNet Guizilini et al. (2020a) in 4 out of 7 metrics by a considerable margin and is comparable for the other three metrics even though the PackNet has 8.5× more parameters than ours. On the other hand, our AQUANet$_{R34}$ outperforms all previous methods in all metrics by a large margin, showing that our method well scales-up with network parameters. Our superior quantitative results are supported by qualitatively more consistent depth estimates, as shown in Fig. 10.

CityScapes. Firstly, we demonstrate that our method benefits from additional data, as shown in the (K+CS) results of Table 2. Secondly, to test the generality of our method, we train our models on (CS) and test them on (K). The consistent improvements are observed in Table 2 from the first training stage (RMSE: 5.396) to the second stage (RMSE: 4.686) regardless of the training data, which implies that our method generalizes well. Additionally, as shown in the second column of Fig. 8, our AQUANet, even with no KITTI images during training, predicts the moving car in the KITTI scene in the first row with infinite depths and the moving cars in the second row with implausible close-by depths. These experimental results imply that *CNN's trained for the SVDE task from videos do not memorize moving objects but rather learn their appearances and potential locations*. More experimental results are provided in the supplementary materials.

5 Conclusions

We have proposed the use of positional information for the first time to aid in self-supervised learning of SVDE from videos. We showed that an SVDE network implicitly learns moving objects' potential locations and appearances. We exploited this behavior to measure depth variances to shifted positional information to compute robust moving object masks, which we call SPIMO masks. We used these masks in our novel learning pipeline, making it possible to remove the moving objects in the photometric reconstruction losses robustly. Additionally, we presented adaptive disparity quantization, utilized by our AQUANet, which in conjunction with the boosting of moving object depths, yields the SOTA results for the self-supervised learning of SVDE from videos.

REFERENCES

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4009–4018, June 2021.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pp. 2366–2374, 2014.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, 2018.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE, 2012.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3828–3838, 2019.

Juan Luis Gonzalez and Munchurl Kim. Plade-net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6851–6860, June 2021.

Juan Luis Gonzalez Bello and Munchurl Kim. Self-supervised deep monocular depth estimation with ambiguity boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3124079.

Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12626–12637. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/951124d4a093eeae83d9726a20295498-Paper.pdf.

Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8977–8986, 2019.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020a.

Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=ByxT7TNFvH.

Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11078–11088, June 2021.

Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7683–7692, 2019.

Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 807–814. IEEE, 2005.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12642–12652, October 2021a.

Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. *CoRR*, abs/2108.08829, 2021b. URL https://arxiv.org/abs/2108.08829.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision*, pp. 582–600. Springer, 2020.

Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2007.

Lina Liu, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Self-supervised monocular depth estimation for all day images using domain separation. *CoRR*, abs/2108.07628, 2021. URL https://arxiv.org/abs/2108.07628.

Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 155–163, 2018.

Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: high resolution self-supervised monocular depth estimation. *CoRR abs/2012.07356*, 2020.

S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9685–9694, June 2021.

Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, pp. 572–588. Springer, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9799–9809, 2019.

Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pp. 11–20. IEEE, 2017.

Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark. *CoRR*, abs/2108.03830, 2021a. URL https://arxiv.org/abs/2108.03830.

Lijun Wang, Yifan Wang, Linzhao Wang, Yunlong Zhan, Ying Wang, and Huchuan Lu. Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12727–12736, October 2021b.

Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2162–2171, 2019.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pp. 842–857. Springer, 2016.

Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Unsupervised high-resolution depth learning from videos with dual networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6872–6881, 2019.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.

Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

A Appendix

A.1 Introduction

In this Appendix, we first provide details on our resize and crop data augmentations specifically tailored for learning depth from videos. We also provide complementary ablation studies and additional visualizations of our proposed adaptive quantization and SPIMO masks. Additionally, we present full versions of Tables 1 and 2 of our main manuscript as well as GIFs, "GIF1-5", that aid in visualizing the generated disparity maps, SPIMO masks, and synthetic views that make learning robust single-view depth estimation (SVDE) from monocular videos possible. Finally, we discuss our currently proposed method's inference times and limitations in these supplemental materials.

A.2 Depth Cue-Preserving Data Augmentation

We propose the use of random resize and random crop data augmentations to train our SVDE network, which are well-known in fully supervised learning. In (GonzalezBello & Kim, 2020; Gonzalez & Kim, 2021), these data augmentations have already been exploited when learning from stereo. However, interestingly and to the best of our knowledge, random resize and crop are commonly avoided in most (if not all) previous works that learn from videos (Zhou et al., 2017; Godard et al., 2019; Guizilini et al., 2020a;b). Applying these data augmentations can be achieved if the changes in scale and cropping coordinates are properly reflected into the camera focal length and principal points, respectively, with a few caveats.

Firstly, the pose estimation network requires to "look" at full-images, with little regard to the image resolutions, to be able to reason about global camera motion. This is not an issue as the final outputs of the PoseNet are just six values per reference camera. So we can feed low-res full images to the PoseNet.

Secondly, we observed that the learning of depth can be controlled in different ways according to the result of the random resizing and cropping operations. Re-scaling the estimated camera translations relative to the random resizing factors can guide the SVDE network to generate disparity estimates that are:

Table 3: Ablation studies of our AQUANet on KITTI (Geiger et al., 2012). Metrics defined in (Eigen et al., 2014). Error metrics are the lower the better and accuracy metrics are the higher the better.

Expmt.	Methods	abs rel	sq rel	rmse	rmse_{log}	δ^1	a^2	a^3
1	AQUANet (1st, W/o PI)	0.118	1.157	4.099	0.164	0.890	0.974	0.988
	AQUANet (1st)	0.120	1.457	4.286	0.167	0.891	0.970	0.986
	AQUANet (1st-stat)	0.080	0.287	2.858	0.114	0.936	0.992	0.999
2	AQUANet (with Godard et al. (2019) masks)	0.113	0.820	4.236	0.167	0.881	0.970	0.989
	DIRNet-BW	0.098	0.579	3.910	0.149	0.900	0.976	0.992
	vPLADE-Net	0.084	0.366	3.240	0.124	0.930	0.987	0.997
	AQUANet	0.080	0.328	3.095	0.118	0.934	0.989	0.997
3	AQUANet($a_b = 0.1$)	0.078	0.320	3.117	0.117	0.935	0.989	0.997
	AQUANet(full)	0.077	0.324	3.032	0.115	0.938	0.989	0.997
	AQUANet_{R34}(full)	0.070	0.285	2.988	0.107	0.946	0.991	0.998
4	AQUANet (no aug.)	0.098	0.532	3.923	0.146	0.902	0.978	0.993
	AQUANet (inverse)	0.091	0.417	3.446	0.131	0.918	0.986	0.996
	AQUANet (invariant)	0.087	0.385	3.288	0.126	0.925	0.987	0.997
5	AQUANet(per-dataset)	0.078	0.332	3.070	0.117	0.935	0.988	0.997
	AQUANet(per-image)	0.085	0.362	3.373	0.128	0.921	0.985	0.997

- **Invariant to image resolutions.** This is the case when the estimated camera translations are not re-scaled by the random scale factor.
- **Inversely proportional.** If the estimated camera translations are multiplied by the resizing scale factor, the SVDE network will learn to generate smaller-valued disparities as the resolution of the image grows. This is, *the larger the world objects are, the farther away they are*. While this might seem to be the most reasonable approach, it breaks one of the most important cues in SVDE: *relative object size* - the closer the object is, the bigger its relative size.
- **Directly proportional.** Dividing the camera translations by the resizing scale factor leads the SVDE model to generate disparity values that scale with image resolutions. This seems to be counter-intuitive, as it could suggest that the larger the world is, the closer the objects are. However, this approach effectively augments the dataset without breaking the relative object size cue.

The benefits of our proposed strategy are in: (i) Flexibility to learn depths at very-high resolutions, (ii) Lower hardware requirements for high-resolution data, and (iii) better generalization as data is effectively augmented. In addition to adopting disparities that scale directly proportional to image resolutions, we also incorporate random horizontal flip, random gamma, random brightness, and random color shifts into our training as in (Godard et al., 2017; GonzalezBello & Kim, 2020).

A.3 Complementary Ablation Studies

Table 3 supplements Table 1 of our main manuscript with experiments 4 and 5 (Expmt. 4 and 5), which explore the effects of our proposed data augmentation protocol and the effects of adaptive quantization, respectively.

Expmt. 4. We ablate the effects of our proposed data augmentation protocol, for AQUANets that are trained without random resizing and cropping augmentations, denoted as AQUANet (no aug.), AQUANets that learn to estimate disparity values that don't scale with image resolutions, denoted as AQUANet (invariant), and AQUANets that learn to estimate disparities that scale inversely proportional to image resolutions, denoted as AQUANet (inverse). Note that the AQUANet in Expmt. 2 learns to predict disparity values that proportionally scale with image resolutions. The networks in Expmt. 4 achieve much lower performance than our AQUANet that learns disparity values that scale proportional to the image resolutions as shown in Table 3. Qualitatively, these AQUANets present blurred object borders and less detailed depth maps as shown in the zoom-boxes in Fig. 11. On the other hand, our AQUANet that learns disparity values that scale proportional to the image resolutions yields more consistent and sharper depth estimates as shown in the last column of Fig. 11. Note that, in addition to obtaining lower performance, the model trained without our augmentation protocol, AQUANet (no aug.), is limited to predicting depth maps at **half resolution**, as it was not exposed to randomly resized and cropped patches during training.

Expmt. 5. To further support the effectiveness of per-pixel adaptive quantization we provide additional ablation studies on per-dataset and per-image adaptive quantization. In Expmt. 5 of Table 3,

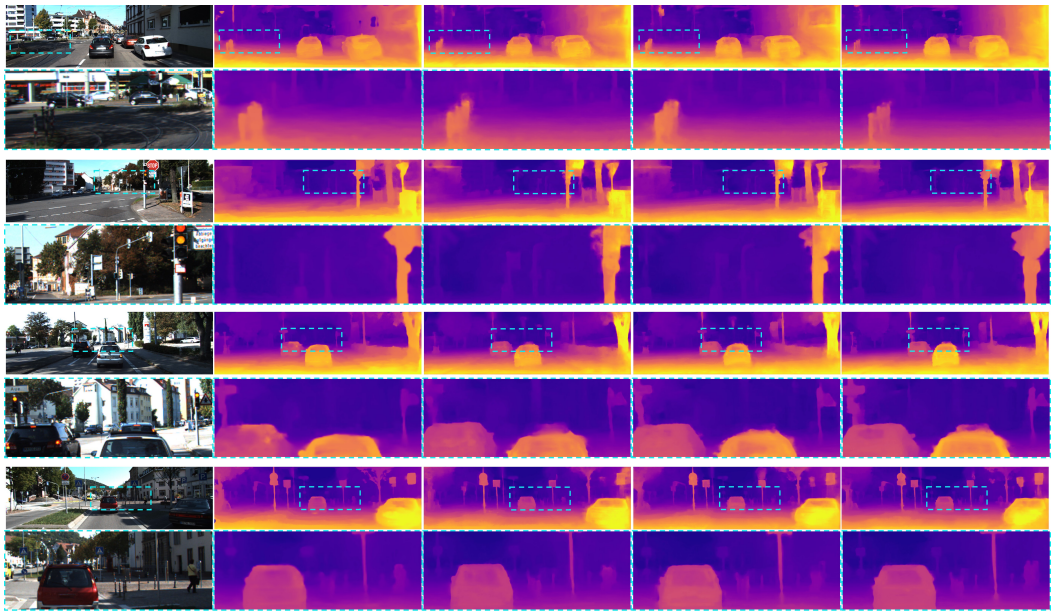

Figure 11: Effects of the proposed data augmentation protocol.

Figure 12: Full images, image crops, and estimated adaptive quantization maps and disparities.

in the AQUANet (per-dataset), we optimize a single adaptive parameter β shared among all image pixels and images in the KITTI dataset. As expected, AQUANet(per-dataset) achieves lower performance than our AQUANet(full) in Expmt. 3 in Table 3, which implemented per-pixel adaptive quantization.

On the other hand, AQUANet(per-image) learns to estimate a single β adaptive parameter per image shared among the image pixels. Interestingly, AQUANet(per-image) yields considerably much lower performance than AQUANet(per-dataset) and AQUANet(full). This is potentially due to the low correlation between the input image and the single adaptive parameter β. In contrast, as depicted in Fig. 12, our adaptive quantization per pixel seems to correlate well to the input image.

A.4 Additional Visualizations

A.4.1 Adaptive quantization

Fig. 12 provides a visualization of the estimated adaptive quantization maps β and disparities by our AQUANet for some training inputs. As described in the Proposed Method Section of our main manuscript, a single channel feature map controls the per-pixel disparity quantization. As can be observed, our adaptive quantization network, the AQUANet, automatically learns to estimate an adaptive quantization map that effectively assigns the most suitable quantization curve for far and

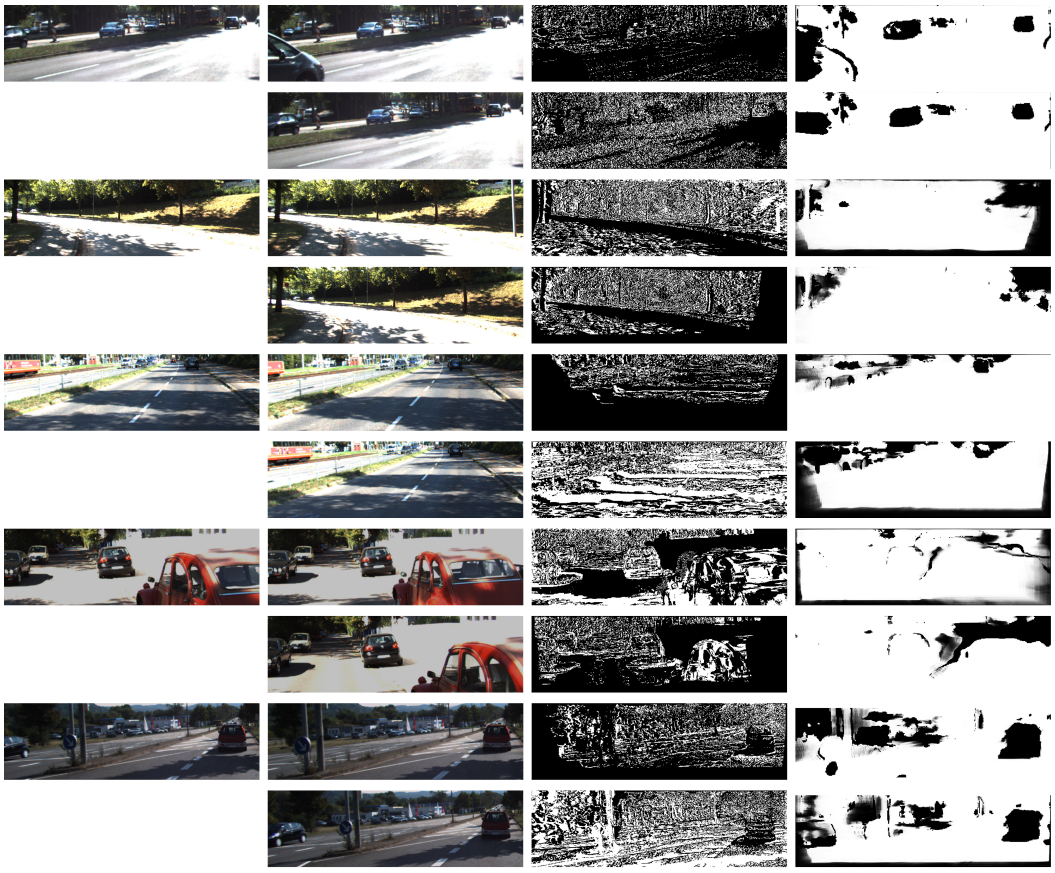

Figure 13: Visual comparison between our shifted positional information moving object (SPIMO) masks and those computed following Monodepth2 (Godard et al., 2019).

close by object pixels. In Fig. 12, lighter pixels denote a quantization curve with more depth levels for distant objects, and darker pixels indicate a quantization curve with more quantization levels for close-by depths.

Interestingly, the adaptive quantization map β is less detailed than the final estimated disparity map, implying that the disparity probability volume (quantized per-pixel by the curves defined by β) is responsible for the fine details in the final depth maps.

Additionally, Fig. 12 depicts the complete images from which random crops are obtained to train our AQUANet. As can be observed, our training strategy allows for learning SVDE from randomly resized and cropped patches.

A.4.2 SPIMO MASKS VS AUTO-MASKS

Figure 13 depicts additional comparisons between the proposed SPIMO masks and their corresponding auto-masks (Godard et al., 2019) for various KITTI (Geiger et al., 2012) sequences. As can be observed, our SPIMO masks are much less sparse and do a better job removing the moving objects in the scene.

A.4.3 SYNTHETIC VIEWS

Our AQUANet learns SVDE via forward warping, that is, by synthesizing the input image seen from the previous and next frames' camera positions. For this reason, it is worthwhile observing those intermediate synthetic views along with their corresponding SPIMO masks and disparities, as depicted in Fig. 14. Even when our AQUANet only learns view synthesis as a pre-text task, the novel views

Figure 14: Additional visualizations of our SPIMO masks, estimated disparities, and synthetic views. Top four rows depict KITTI (Geiger et al., 2012) samples and bottom four rows depict CityScapes (Cordts et al., 2016) samples.

are photo-realistic. Again, our SPIMO masks demonstrate to be very robust and capable of removing the moving objects in the scene. Finally, it is worth noting that the boosted disparities(obtained by loosely following (Gonzalez Bello & Kim, 2021)) are slightly more consistent than the network estimated disparities, which allows for a progressive improvement during training.

To ease the visualization of Figure 14, we provide a GIF, "GIF1", in the supplementary materials. We also provide "GIF2-5" which show synthesized novel views from the KITTI Eigen Test Split Eigen et al. (2014). Note that even when these views are generated from a single image input, they display realistic appearances.

A.5 Full Results Table

Table 4 supplements Table 2 in our main manuscript by adding other recent works for quantitative comparison. Again, our *AQUANet* yields the best RMSE and sq rel in both the original (Eigen et al., 2014) and improved (Uhrig et al., 2017) Eigen test splits when trained either on KITTI (K) or KITTI+CityScapes (K+CS). Note that our proposed method yields the state-of-the-art result even without the use of semantics (as in (Guizilini et al., 2020b; Jung et al., 2021a)) and without exploding number of network parameters (as in Guizilini et al. (2020a)).

Table 4: Comparison to Existing SVDE Methods on the KITTI Eigen Split (Eigen et al., 2014). DoF: depth of field supervision. D: Fully-supervised. V, V_{se}, S, and S_{SGM}: Self-supervised from video, video+semantics, stereo, and stereo+SGM. Models train on KITTI (K) or CityScapes (CS). PP: post-processing. Par: # of params. V models use median scaling.

	Reference	Methods	Sup	Data	Par	abs rel	sq rel	rmse	$\log_{rmse}$	a^1	a^2	a^3
Original Eigen Test Split Geiger et al. (2012)	Luo et al. (2018)	Luo *et al.*	D+S	K	-	**0.094**	**0.626**	4.252	0.177	**0.891**	0.965	0.984
	Gur & Wolf (2019)	Gur *et al.*	DoF	K	-	0.110	0.666	**4.186**	**0.168**	0.880	**0.966**	**0.988**
	Watson et al. (2019)	DepthHints(PP)	S_{SGM}	K	35	0.096	0.710	4.393	0.185	0.890	0.962	0.981
	GonzalezBello & Kim (2020)	FAL-net	S	K+CS	17	0.091	**0.562**	4.016	0.178	0.894	0.964	0.983
	Gonzalez & Kim (2021)	PLADE-Net	S	K+CS	15	**0.090**	0.577	**3.880**	**0.170**	**0.903**	**0.968**	**0.985**
	Gordon et al. (2019)	Gordon *et al.*	V	K	-	0.128	0.959	5.230	0.212	0.845	0.947	0.976
	Zhou et al. (2019)	Zhou *et al.*	V	K	34	0.121	0.837	4.945	0.197	0.853	0.955	0.982
	Godard et al. (2019)	Monodepth2	V	K	14	0.115	0.882	4.701	0.190	0.879	0.961	0.982
	Shu et al. (2020)	FeatDepth	V	K	73	0.104	0.729	4.481	0.179	0.893	0.965	**0.984**
	Guizilini et al. (2020a)	PackNet	V	K	120	0.107	0.802	4.538	0.186	0.889	0.962	0.981
	Lyu et al. (2020)	HR-Depth	V	K	14	0.104	0.727	4.410	0.179	0.894	**0.966**	**0.984**
	Jung et al. (2021a)	FSRE-Depth (R18)	V_{se}	K	25	0.105	0.722	4.547	0.182	0.886	0.964	**0.984**
	Jung et al. (2021a)	FSRE-Depth (R50)	V_{se}	K	25	0.102	0.675	4.393	0.178	0.893	**0.966**	**0.984**
	Proposed	*AQUANet*	V	K	14	0.115	0.656	4.251	0.186	0.875	0.959	0.983
	Proposed	$AQUANet_{R34}$	V	K	25	0.105	**0.621**	4.227	0.179	0.889	0.964	**0.984**
	Gordon et al. (2019)	Gordon *et al.*	V	K+CS	-	0.124	0.930	5.120	0.206	0.851	0.950	0.978
	Klingner et al. (2020)	SG-Depth	V	K+CS	16	0.107	0.768	4.468	0.186	0.891	0.963	0.982
	Guizilini et al. (2020a)	PackNet	V	CS→K	120	0.104	0.758	4.386	0.182	0.895	0.964	0.982
	Guizilini et al. (2020b)	Guizilini *et al.*	V_{se}	CS→K	140	**0.100**	0.761	4.270	**0.175**	**0.902**	0.965	0.982
	Proposed	*AQUANet*	V	K+CS	14	0.112	0.669	4.263	0.182	0.882	0.963	**0.984**
	Proposed	$AQUANet_{R34}$	V	K+CS	25	0.108	0.649	**4.224**	0.181	0.890	0.964	**0.984**
	Proposed	*AQUANet*(PP)	V	K+CS	14	0.110	0.637	4.170	0.179	0.885	0.964	0.984
	Proposed	$AQUANet_{R34}$(PP)	V	K+CS	14	0.103	0.601	4.137	0.176	0.892	0.965	0.984
Improved Test Split Uhrig et al. (2017)	Yin et al. (2019)	Yin *et al.*	D	K	113	0.072	-	3.258	0.117	0.938	0.990	0.998
	Fu et al. (2018)	DORN	D	K	51	0.072	0.307	2.727	0.120	0.932	0.984	0.995
	Guizilini et al. (2021)	PackNet-SAN	D	K	>120	**0.052**	**0.175**	**2.233**	**0.083**	**0.970**	**0.996**	**0.999**
	Godard et al. (2019)	Monodepth2	V+S	K	14	0.087	0.479	3.595	0.131	0.916	0.984	0.996
	Watson et al. (2019)	DepthHints(PP)	S_{SGM}	K	35	0.074	0.364	3.202	0.114	0.936	0.989	0.997
	GonzalezBello & Kim (2020)	FAL-net(PP)	S	K+CS	17	0.068	0.276	2.906	0.106	0.944	0.991	**0.998**
	Gonzalez & Kim (2021)	PLADE-Net	S	K+CS	15	**0.066**	**0.263**	**2.726**	**0.102**	**0.949**	**0.992**	**0.998**
	Proposed	$AQUANet(1^{st})$	V	CS	14	0.179	1.881	5.396	0.240	0.790	0.933	0.969
	Proposed	*AQUANet*	V	CS	14	0.131	0.786	4.686	0.186	0.837	0.962	0.988
	Godard et al. (2019)	Monodepth2	V	K	14	0.092	0.536	3.749	0.135	0.916	0.984	0.995
	Jung et al. (2021a)	FSRE-Depth (R18)	V_{se}	K	25	0.084	0.436	3.740	0.129	0.919	0.985	0.996
	Guizilini et al. (2020a)	PackNet	V	K	120	0.078	0.420	3.485	0.121	0.931	0.986	0.996
	Proposed	$AQUANet(1^{st})$	V	K	14	0.120	1.457	4.286	0.167	0.891	0.970	0.986
	Proposed	*AQUANet*	V	K	14	0.077	0.324	3.032	0.115	0.938	0.989	0.997
	Proposed	$AQUANet_{R34}$	V	K	25	**0.070**	0.285	2.988	**0.107**	0.946	0.991	**0.998**
	Guizilini et al. (2020a)	PackNet	V+v	CS→K	120	0.075	0.384	3.293	0.114	0.938	0.984	0.995
	Guizilini et al. (2020a)	PackNet	V	CS→K	120	0.071	0.359	3.153	**0.109**	0.944	0.990	0.997
	Proposed	$AQUANet(1^{st})$	V	K+CS	14	0.114	1.171	4.055	0.162	0.898	0.975	0.988
	Proposed	*AQUANet*	V	K+CS	14	0.076	0.301	**2.921**	0.111	0.943	0.991	**0.998**
	Proposed	$AQUANet_{R34}$	V	K+CS	25	**0.070**	**0.283**	2.943	**0.107**	**0.948**	**0.992**	**0.998**
	Proposed	*AQUANet*(PP)	V	K+CS	14	0.073	0.279	2.813	0.107	0.946	0.992	0.998
	Proposed	$AQUANet_{R34}$(PP)	V	K+CS	25	0.067	0.262	2.844	0.102	0.951	0.993	0.998

It is worth noting that we provide results for our AQUANet with post-processing, denoted as (PP) in Table 4. In this case, the post-processing is the boosting process detailed in Section 3.4 of our main manuscript. The slightly higher performance of the AQUANets with (PP) allows for incremental improvements in the moving object regions when employed in the boosting loss, as described in Sections 3.4 and 3.5 of our main manuscript.

A.6 Run-time analysis

Our 14 million parameters AQUANet sits on the lower spectrum for depth estimation networks regarding the number of parameters. In terms of inference times, our AQUANet is comparable to other light depth estimation networks (Godard et al., 2017; 2019). Our network performs disparity inference of a full KITTI image of 1242×375 pixels in 66ms, and 22ms for a half-resolution

KITTI image of 621×187 resolution on the vanilla PyTorch on a Titan XP GPU, leaving room for improvements when implementing real-time libraries such as TensorRT.

When performing view synthesis on a full KITTI image, our AQUANet has a latency of 90ms or 33ms for half-resolution images. This is due to the additional computations (e.g., $2N$ warping, softmax, and dot-product operations, where N is the number of quantization levels) needed to perform free-view synthesis, as described in Section 3.2 of our main manuscript.

A.7 LIMITATIONS

Although our SPIMO masks effectively remove moving objects in the scene, their computation is easily affected by γ in Eq. 7 of our main manuscript. We empirically demonstrated the handcrafted threshold of $\gamma = 3\%$, but this might not be optimal for all image cases. Thus, exploring adaptive thresholds for SPIMO masks is an interesting problem for future work. Furthermore, the additional fixed-network forward passes needed to compute the depth variances to shifted positional information lead to an increased training time, which could be considered a limitation. However, this additional learning time can be eliminated if the moving object masks are pre-computed before the second training stage.