Windowed Bundle Adjustment Framework for Unsupervised Learning of Monocular Depth Estimation with U-Net Extension and Clip Loss

Lipu Zhou and Michael Kaess

***Abstract*— This paper presents a self-supervised framework for learning depth from monocular videos. In particular, the main contributions of this paper include: (1) We present a windowed bundle adjustment framework to train the network. Compared to most previous works that only consider constraints from consecutive frames, our framework increases the camera baseline and introduces more constraints to avoid overfitting. (2) We extend the widely used U-Net architecture by applying a Spatial Pyramid Net (SPN) and a Super Resolution Net (SRN). The SPN fuses information from an image spatial pyramid for the depth estimation, which addresses the context information attenuation problem of the original U-Net. The SRN learns to estimate a high resolution depth map from a low resolution image, which can benefit the recovery of details. (3) We adopt a clip loss function to handle moving objects and occlusions that were solved by designing complicated network or requiring extra information (such as segmentation mask [1]) in previous works. Experimental results show that our algorithm provides state-of-the-art results on the KITTI benchmark.**

I. Introduction

Predicting depth from a single image is challenging. However, it has many applications in 3D vision and robotics, such as autonomous driving, obstacle avoidance and Simultaneous localization and mapping (SLAM). Due to its importance, much effort has been devoted towards solving this problems. Early works [2], [3], [4] formulated this task as a supervised learning problem. The difficulty of the supervised method lies in the lack of ground truth depths. Recent work [5], [6], [7], [8], [9], [10], [1] shows that view synthesis provides an effective supervisory signal to train the network, and this makes unsupervised learning for depth estimation possible.

In the literature, stereo and monocular video has been used to train the network. Compared to stereo video, monocular video is a larger training source. However, training on monocular videos is more challenging, due to the unknown camera motion, moving objects, and varying lighting conditions. This paper focuses on unsupervised learning using monocular video.

Typically, generating effective geometric constraints, designing proper network architecture and handling moving objects and occlusion are important for learning high quality depth estimation. It is well known in the field of multiple view geometry [11] that a large camera baseline is essential for accurate depth estimation. But most of previous self-supervised frameworks [5], [12], [7], [13], [1], [14] only consider consecutive frames to generate constraints. The small baseline between two consecutive frames may reduce the depth estimation accuracy.

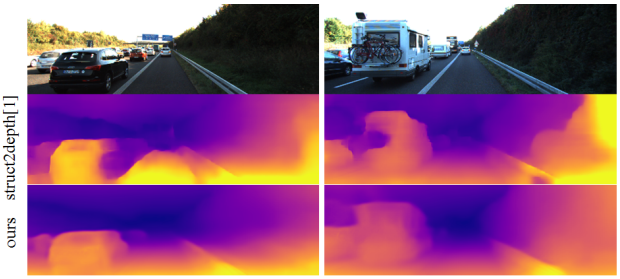

Fig. 1. Results of our algorithm and strut2depth [1]. Strut2depth [1] uses segmentation mask and pretrained model for training. Our algorithm does not require extra data and learns from scratch, but it provides better results

In terms of the network architecture, the U-Net [15] is generally adopted for depth estimation. The U-Net has a decoder-encoder structure with skip connections to combine feature maps of different size of receptive field as shown in Fig. 3 (a). Context information is important for depth estimation as demonstrated in Fig. 2. However, in the U-Net structure [15], the context information from the encoder may be gradually attenuated as it is sequentially combined with feature maps with diminishing receptive fields. This may make the depth estimation sensitive to local texture. Besides, previous work has generally adopted the paradigm of producing a low resolution depth map from a low resolution image, followed by a up-sampling step. The interpolation may result in a blurry depth map and ignore details. The above issues are ignored by previous work. Moreover, one challenge for learning depth from monocular video lies in moving objects and occlusions. Previous works generally design complicated networks or require extra information to solve this problem, such as the expandability net designed in [5] or the segmentation mask required in [1]. This paper aims to solve the above issues. Experimental results show that our solution yields state-of-the-art results on the KITTI dataset. The main contributions of this paper are the following:

Windowed Bundle Adjustment Framework We present a Windowed Bundle Adjustment Framework (WBAF) to train the network. Our WBAF jointly optimizes depths and camera poses through cross-sequence photometric and geometric constraints in forward and backward directions as shown in Fig. 4. Compared to previous work, our framework increases the baseline and introduces more constraints to avoid overfitting.

U-Net Extension We extend the U-Net by applying a Spatial Pyramid Net (SPN) and a Super-Resolution Net

This work was partially supported by the Office of Naval Research under grant N00014-16-1-2103.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA zhoulipu@outlook.com, kaess@cmu.edu

(SRN). Motivated by [16], our SPN uses skip connects to fuse information from regions in a pyramid, which can provide different levels of context information for depth estimation. Our SRN learns to generate a high resolution depth map from a low resolution input.

Clip Loss Function We adopt a clip loss function to deal with moving objects and occlusion. During training, errors higher than a certain percentile will be capped. They will generate zero gradients, and will not impact on training.

Some recent work [17], [18], [19] presented similar concepts as our algorithm. But our algorithm is different from them. Our WBAF differs from the BA-Net introduced in [17]. BA-Net is a supervised framework whose purpose is yield a differentiable Levenberg-Marquardt (LM) algorithm. However, our WBAF is an unsupervised framework that aims to increase the camera baseline and introduce more constraints. Our SPN is also different from the feature pyramid introduced in [17]. The feature pyramid [17] is trained to replace the image photometric error and is not used to predict the depth. Our SPN, on the other hand, is specifically designed to predict the depth. Our SPN also differs from the pyramid pooling model [18] that is not of an encoder-decoder structure. It applies average or maximum pooling to the last layer of a CNN to generate feature maps with different scale. Our SPN has an encoder-decoder structure and combines feature maps from different convolutional layers. Our SRN also differs from SuperDepth [19] where the sub-pixel convolution [20] is adopted to replace the up-sampling layers in the U-Net. Their input and output are of the same size. But our SRN is used to generate a high resolution depth map from a low resolution input. Our SRN can provides 4 times more constraints than SuperDepth.

A. Related Work

Estimating depth from a single image is a challenging task. A large number of supervised and unsupervised approaches have been proposed to address this problem.

Supervised Depth Estimation Most supervised approaches formulate the depth estimation problem as a supervised regression problem. In early work [21], a Markov random field (MRF) with hand-crafted features was trained to estimate the depth. To avoid feature engineering, Eigen *et al.* [22] presented a multi-scale CNN to predict the depth. Recently, Tang *et al.* [17] introduced BA-Net and feature pyramid to improve the performance. Nekrasov *et al.* [23] combined semantic segmentation and depth estimation into one model and achieved real-time performance. Li *et al.* [24] employed structure-from-motion (SfM) and multi-view stereo (MVS) technologies to generate the supervisory 3D information. But this method is not applicable to scenarios where SfM or MVS fail to work.

Unsupervised Depth Estimation The photometric and geometrical consistency of nearby frames provides a way to avoid the requirement of ground truth depths. Stereo and monocular sequences are used to train the network.

Stereo Sequences The left and right images and the known pose between them form a self-supervisory loop to train the network. Garg *et al.* [25] first applied this self-supervised methodology on stereo sequences. They used the Taylor expansion to approximate the cost function for gradient computation, which may result in a suboptimal objective. To solve this problem, Godard *et al.* [6] applied the spatial transformer network [26] to yield a differentiable reconstruction cost function. The temporal photometric and deep feature reconstruction errors were used to improve the performance in [8]. Aleotti *et al.* [27] designed a GAN paradigm [28] for the depth learning. Geo *et al.* [29] employed a stereo matching network to supervise the depth learning. Recently, Zhan *et al.* [14] showed improved accuracy by using the depth-normal consistency to train two networks for depth and normals estimation.

Monocular Sequences Learning depth from monocular sequences is more challenging, due to the unknown camera pose and moving objects. Zhou *et al.* [5] showed that it is capable of learning depth and pose estimation at the same time. Several recent works explore additional geometrical constraints for the unsupervised training. The consistency between normals and depths was utilized in [10]. The 3D point cloud alignment loss was introduced in [13]. Depth and optical flow predictions are related tasks. Recent works [12], [7] showed that jointly learning them can be of mutual benefit. Motivated by traditional direct visual odometry (DVO) technology, Wang *et al.* [9] introduced depth normalization and a differentiable DVO module to replace the pose network. Recently, Godard *et al.* [30] presented several effective approaches to improve the depth estimation.

Previous work generally only consider the constraints from two consecutive frames, which results in a short baseline. Besides, the generally used U-Net structure [15] has the risk of context information reduction. Additionally, the interpolation used to recover a high resolution depth map may result in a blurry depth map. This paper aims to address these issues.

Moving objects and occlusion are another problem for unsupervised training. Zhou *et al.* [5] proposed an explanation mask to estimate the regions undergoing motion and occlusion. However, they later found that the explanation mask actually reduced the performance. In [7], [12], [31], a separate optical flow model was trained to handle moving objects. Prasad *et al.* [32] computed the essential matrix to deal with moving objects. Recently, Casser *et al.* [1] exploited the segmentation mask to model object motion. The central idea of addressing moving objects is to eliminate them from training. Thus, this paper introduces a simple method, *i.e.* a clip loss function, to solve this problem.

II. Our Approach

Our framework includes two networks for depth and pose estimation, as demonstrated in Fig. 4. We will detail our unsupervised framework below.

A. Depth Net

This section describes the architecture of our depth net. It extends the U-Net with SPN and SRN.

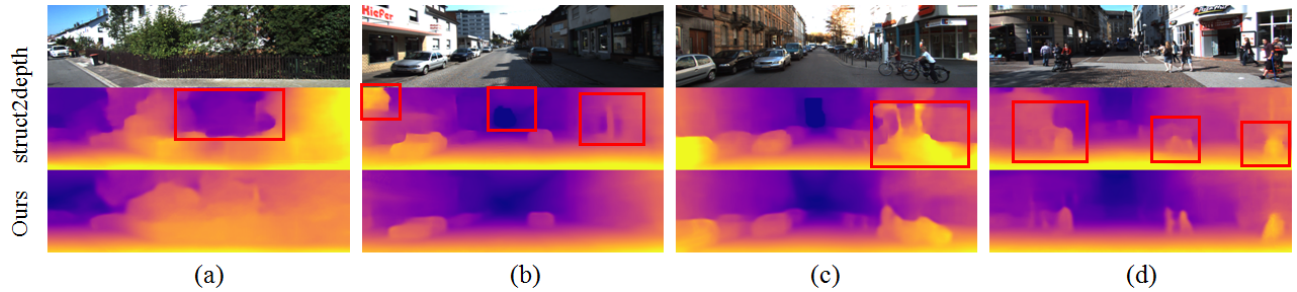

Fig. 2. Motivations for our spatial pyramid net. Context information provides essential information for depth inference. Actually, even our human can not infer the depth of a pixel from a small path around it for some objects. However, in the traditional U-Net [15], the context information is gradually attenuated as it sequentially combines with features with reducing receptive fields, which may make it sensitive to local texture as shown in (a) and (b). Besides, local features will benefit recovering the details of complicated objects, such as the two boys riding bicycles in (c) and the pedestrians in (d). Therefore, context and local information are both important for depth estimation. Our algorithm combines information from a spatial pyramid.

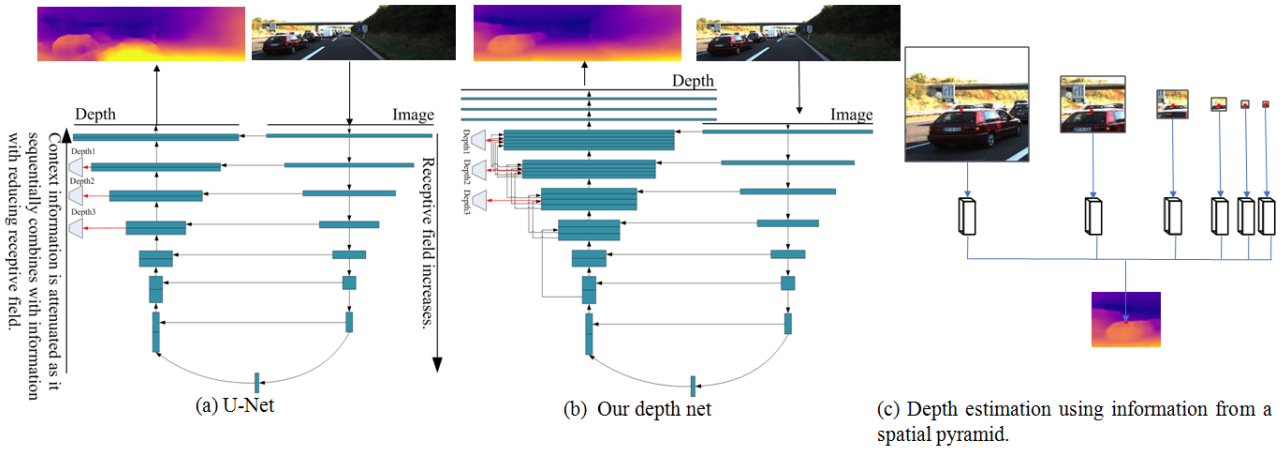

Fig. 3. **(a) A schematic of the U-Net.** In the U-Net, feature maps from the encoder with decreasing receptive fields are sequentially infused into the decoder. This may downplay the context information. **(b) A schematic of our U-Net extension including SPN and SRN.** The SPN fuses feature maps with different size of receptive field by skip connections. The SRN has 3 layers and generates a high resolution depth map. **(c) The effect of our SPN.** Our SPN integrates nformation from an image pyramid to estimate the depth.

1) Motivations for SPN: In previous work, the U-Net architecture [15] is adopted to predict the depth. The U-Net includes an encoder and a decoder, as demonstrated in Fig. 3 (a). The last feature map $\mathbf{FM}_N$ of the encoder has the largest receptive field, and is the input of the decoder. As $\mathbf{FM}_N$ goes through the decoder, feature maps with decreasing receptive fields from lower layers of the encoder sequentially merge into it. As the local information accumulates, context information gradually reduces. But the context information is important for depth estimation due to the following reasons:

(I) Context information can increase the consistency of the depth map. Nearby pixels have similar context information which can avoid anomalous depth estimation caused by variation in local appearance. As shown in Fig. 2 (a), the state-of-the-art algorithm [1] generates a hole in the bush labeled by a red box. This may be cased by the mixture of the vegetation and the background white wall.

(II) Context information provides semantic information to assist in inferring depth in complex situations. For example, the depth of the letters and windows on the wall in Fig. 2 (b) should be consistent with the depth of the wall. Their depth can not be correctly inferred without the wall as a reference.

(III) Context information is essential to distinguish pixels in textureless regions that are locally similar, but globally different. For example, as shown in Fig. 2 (b), even our human requires the context information to distinguish the pixel of the wall and the pixel of the sky.

On the other hand, local information is important to recover the details, such as object boundaries. For instance, Fig. 2 (c) and (d) include persons in various poses. The local information is essential to recover their depth.

2) Spatial Pyramid Net: As described above, both local and context information are important for depth estimation. Therefore, we present a SPN to fuse information from an image pyramid, as demonstrated in Fig. 3. Our SPN uses skip connections to concatenate feature maps with different size of receptive field for depth estimation. We call the feature map generated in this way as the Spatial Pyramid Feature Map (SPFM). As previous works, apart from the main branch, our SPN has three auxiliary depth predictors. For each predictor, the current finest feature map is combined with previous coarser feature maps to yield a SPFM for depth estimation. As a finer feature map has a smaller number of channels, the number of channels of a coarser feature map is reduced by half for each skip.

3) Super-Resolution Network: In previous work, the network takes a down-sampled image as the input, and predicts a depth map of the same resolution. This low resolution depth map is then interpolated to recover the original resolution. The drawback of this method is that it may yield a blurry depth map. Furthermore, details or even entire small objects may be lost. To solve this problem, we introduce a SRN

which is trained to generate a depth map of double the size compared to the input image. Therefore, our SRN provides 4 times as many training constraints compared to previous works. This can benefit the training. Our SRN upsamples the last feature map of our SPN, followed by three convolutional layers for depth prediction. Fig. 3 (b) demonstrates the structure of our SRN. From Fig. 6, we can find that our network can generate clear boundaries and more details of the scene.

4) Depth Net Architecture: Our depth net adopts the architecture of [5] as the backbone. Our SPN introduces some skip connections to concatenate feature maps with different size of receptive field. Each skip connection uses a 3×3 kernel to halve the number of channels, then upsampling is adopted to double the size of the feature map. Our SRN includes three layers using a 3×3 kernel with the number of channels 16, 8 and 8, respectively. We adopt the depth map normalization as in [9]. We apply the sigmoid function at the depth estimation, and the ReLU non-linearity elsewhere.

B. Pose Net

Our pose net has a similar structure as [5], except that our pose net takes two consecutive images as the input, and predicts the relative pose between them. As the pose net sequentially slides through the whole sequence, it estimates the relative pose of each image pair. This differs from [5], where the whole image sequence is fed into the pose network, and the poses between the middle image and others are predicted. One advantage of our pose net is that it can be trained on short sequences, but applied to a sequence with arbitrary length.

C. Windowed Bundle Adjustment Training Framework

In the traditional DVO system, landmarks are tracked by photometric consistency frame by frame. The Windowed Bundle Adjustment Framework (WBAF) is performed to jointly optimize a set of camera poses and landmark depths within a sliding window [33]. The WBAF establishes the constraints between non-consecutive camera poses. This increases the baseline of a moving camera, which is essential for accurate depth estimation. Motivated by this, we formulated our unsupervised training in the WBAF, as demonstrated in Fig. 4. Our WBAF uses N-frame snippet $S = \{I_1, I_2, \cdots, I_N\}$ as the input. Similar to the traditional WBAF, the photometric consistency is used to track each pixel frame by frame. Besides, the depth consistency is exploited to establish the *cross-sequence geometric constraints* on camera poses and depths. Specifically, we consider the depth consistency between one depth map and all the remaining ones. Then a set of poses and depth maps are jointly optimized. Apart from the forward motion, we also consider a backward motion which reverses the sequence during the training. Our WBAF establishes the constraints among the N frames. Compared to previous work [5], [12], [7], [1], [14] that only considers constraints from two consecutive frames, our frame yields more constraints to avoid overfitting. Specifically, our algorithm generates the number of constraints of the order $O(N^2)$, instead of $O(N)$ in previous work. Our WBAF minimizes the photometric consistency cost, geometric consistency cost and local smoothness cost detailed below.

1) Photometric Consistency Cost: We first consider the photometric cost from 2 images I_t and I_{t+1}. Given the estimated depth map $\hat{D}_t$ from I_t and the estimated pose $\hat{T}_{t\to t+1}$ between I_t and I_{t+1}, we can map a homogeneous pixel $p_t \in I_t$ onto a pixel $\hat{p}_{t+1} \in I_{t+1}$ as

$$\hat{p}_{t+1} \sim K\hat{T}_{t\to t+1}\hat{D}_t\left(p_t\right)K^{-1}p_t \tag{1}$$

where K is the camera intrinsic matrix. As previous work [8], we adopt the differentiable spatial transformer network introduced in [26] to calculate the value of $I_{t+1}\left(\hat{p}_{t+1}\right)$. Specifically, $I_{t+1}\left(\hat{p}_{t+1}\right)$ is calculated by the bilinear interpolation using the values of the 4 neighbors around $\hat{p}_{t+n}$. Using this method, we can reconstruct I_t by I_{t+1} and $\hat{T}_{t\to t+1}$. Assuming a static scene, no occlusion, and constant lighting conditions, $\hat{I}_t$ is expected to be the same as I_t. As some pixels of I_t may not be visible in I_{t+1}, we use the mask $M_t\left(p_t\right)$ proposed in [13] to get rid of these invisible pixels. We formulate the photometric consistency cost between $\hat{I}_t$ and I_t as

$$\begin{aligned} &L_{re} = \sum_{t=1}^{N-1}\sum_{p_t} M_t(p_t)L_{re}^t\left(p_t\right), \\ &L_{re}^t\left(p_t\right) = \alpha\tfrac{1-\mathrm{SSIM}(\hat{I}_t(p_t),I_t(p_t))}{2} + (1-\alpha)\left|\hat{I}_t(p_t) - I_t(p_t)\right| \end{aligned} \tag{2}$$

where SSIM represents the structural similarity index [34] and α is set to 0.85 as previous work.

2) Cross-sequence Geometric Consistency Cost: The depth of a 3D point estimated from different images should be consistent. This can be used to establish constraints among images in S. **However, the geometric consistency cost can not be formated as done in [7], using the above photometric consistency cost.** This is because we can assume the color of a 3D point is the same for different frames, however, the depth of a 3D point changes.

For each $p_t \in I_t$, we can use (1) to estimate the corresponding $\hat{p}_{t+n} \in I_{t+n}$. Since $\hat{p}_{t+n}$ has continuous coordinates, we estimate the depth of $\hat{p}_{t+n}$ using the bilinear interpolation. We denote the depth map generated in this way as $\hat{D}_{t\to t+n}\left(p_t\right)$. On the other hand, we can transform the point cloud in frame t to frame $t+n$ using

$$P_{t\to t+n} = \hat{T}_{t\to t+n}\hat{D}_t\left(p_t\right)K^{-1}p_t \tag{3}$$

Then the depth of p_t in frame $t+n$ is the z-coordinate of $P_{t\to t+n}$. We denote the depth map generated from (3) as $\tilde{D}_{t\to t+n}\left(p_t\right)$. Ideally, $\tilde{D}_{t\to t+n}\left(p_t\right)$ and $\hat{D}_{t\to t+n}\left(p\right)$ should be equal. Thus, we have the depth consistency cost as

$$L_{dc} = \sum_{t=1}^{N-1}\sum_{n=1}^{N-t}\sum_{p_t}\left|\tilde{D}_{t\to t+n}(p_t) - \hat{D}_{t\to t+n}(p_t)\right| \tag{4}$$

L_{dc} establishes cross-sequence constraints, which increases the camera base line, as illustrated in Fig. 4.

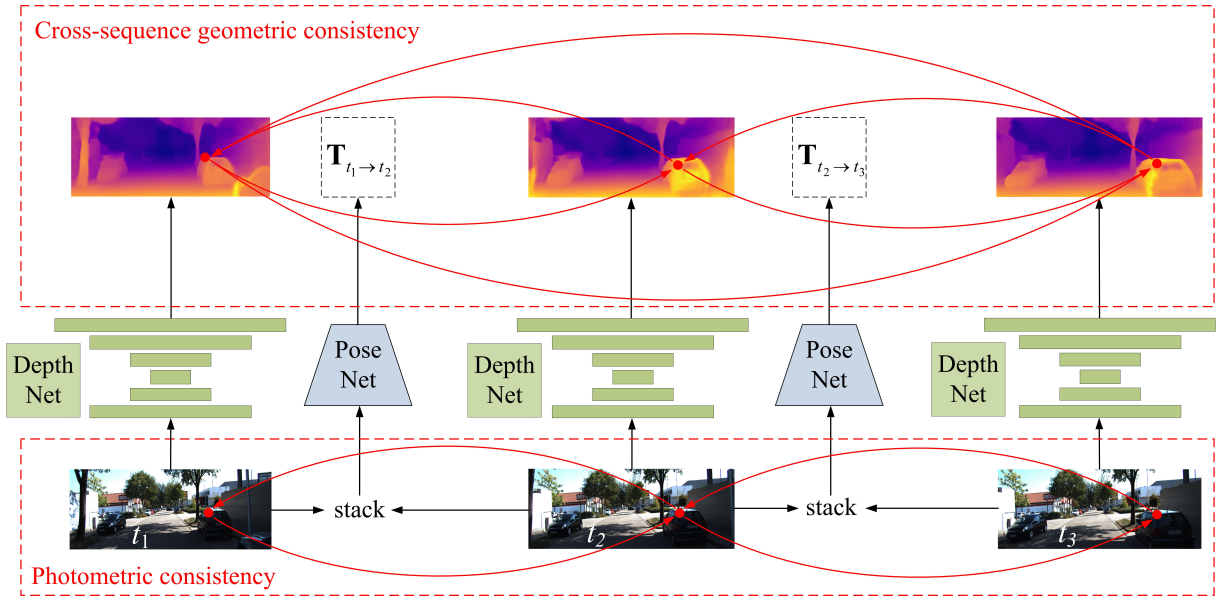

Fig. 4. Our windowed bundle adjustment training framework. Our framework can be applied to a N-frame snippet. Here we use 3 frames as an example. Arcs represent constraints. Previous works [5], [10], [12], [7], [13], [14] typically only consider two consecutive cameras, which results in a short baseline. Motivated by the traditional WBAF, our algorithm jointly optimized the camera poses and depth maps within a sequence. Compared to previous training frameworks, our WBAF increases the baseline and generate more constraints.

Fig. 5. Moving objects generate large errors during training.

3) Spatial Smoothness Cost: The above cost functions are not sufficient to predict the depth of textureless regions. To handle this problem, we adopt the edge-aware smoothness regularization term to encourage local smoothness while allowing sharpness at the edges [6]. As the values for depth are unbounded, we impose the following regularization term on disparity (inverse depth)

$$L_{ds} = \sum_{t=1}^{N} \sum_{i,j} \left|\partial_x d_{ij}^t\right| e^{-\left|\partial_x I_{ij}^t\right|} + \left|\partial_y d_{ij}^t\right| e^{-\left|\partial_y I_{ij}^t\right|} \quad (5)$$

where ∂_x and ∂_y represent the gradient in x and y directions.

4) Backward Sequence: Apart from the general forward sequence, we also reverse the order of the sequence to generate a backward sequence. We construct the cost of the backward sequence in the same manner as the forward sequence. This leads to more constraints that help avoid overfitting. During training, we jointly optimize the forward and backward losses.

D. Clip Loss Function

The above model assumes a static scene and no occlusion. An image region that violates the above assumption will generate a large cost as demonstrated in Fig. 5, and will in turn yield a large gradient that potentially worsens the performance. We treat these violations as outliers, and present a clip function to handle them. Assume c_i is the ith cost in the cost set $\mathbf{C}$. To handle the above problem, we introduce the following robust loss function

$$\rho\left(c_i\right) = \min\left(c_i, \alpha\right), \alpha = p\left(\mathbf{C}, q\right) \quad (6)$$

where $p\left(\mathbf{C}, q\right)$ represents the qth percentile of $\mathbf{C}$. That is to say the cost in $\mathbf{C}$ is clipped at the qth percentile. Costs above the qth percentile will yield zero gradient, and do not affect the training. We apply (6) to the cost functions (2) and (4) introduced above.

E. Total Objective Function

Our learning objective combines the above-mentioned loss functions including both forward and backward sequences. For the cost function (2) and (4), we apply (6) to deal with the moving object and occlusion. We adopt multiple scale losses to train the network. 4 scales are used in the experiments as was done in previous work. The final objective function is

$$L = \sum_{s=1}^{4} \frac{L_s}{2^{s-1}}, L_s = \rho\left(L_{re}^s\right) + \alpha\rho\left(L_{dc}^s\right) + \beta L_{ds}, \quad (7)$$

where $\rho\left(\cdot\right)$ is the clip loss function defined in (6).

III. Experiments

In this section, we compare our algorithm with the state-of-the-art methods. In addition, we conduct the ablation experiment to show that our WBAF, SPN, SRN and clip loss function all benefit depth prediction.

A. Training Details

Our network is implemented using TensorFlow 1.9 and is trained from scratch. We employed the Adam [35] optimizer to minimize the objective function (7) with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The model is trained for 15 epochs with initial

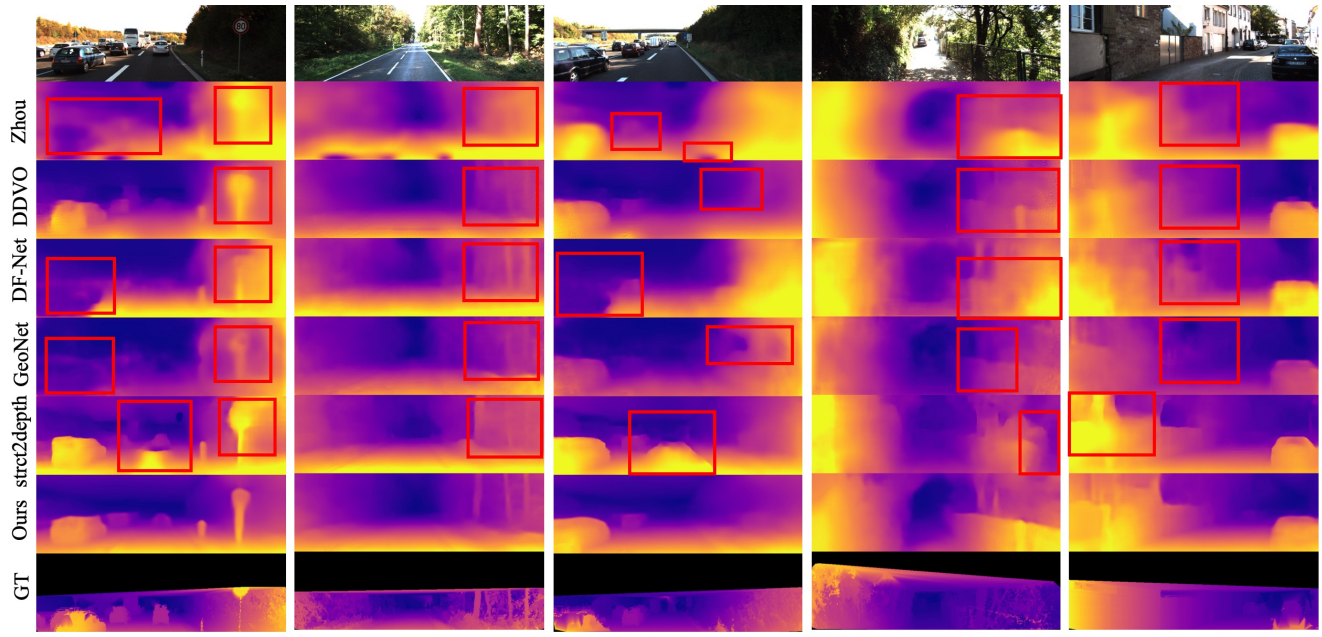

Fig. 6. Qualitative results on the KITTI dataset. The results from top to bottom are from Zhou [5], DDVO [9], DF-Net [7], GeoNet [12], strcut2depth [1] and ours algorithm, respectively. The compared algorithms produce some inaccurate depth estimation, as illustrated by the red rectangles.

learning rate of 10^{-4} for the first 10 epochs, then dropped to 10^{-5} for the last 5 epochs. The weights in (7) are set as $\alpha = 1$ and $\beta = 0.01$ throughout all the experiments below. For the clip loss function (6), we set the percentile $q = 95$ in our experiments. During training, we randomly scale the image contrast with $[0.8, 1.2]$ and jet the brightness with ± 10. The image is resized to 128×416 during training, which results in a 256×832 depth map.

B. Monocular Depth Estimation

We mainly use the KITTI [36] and Make3D [21] datasets to evaluate our algorithm. As previous work, we set the length of the training sequence to 3. The dataset is split as in [22], which generates 40K training samples, 4k evaluation samples, and 697 test samples. The performance is assessed by absolute relative difference, square related difference, RMSE and log RMSE as previous work.

KITTI Table I lists the results of different algorithms. The table is split into several parts according to the supervision level, the dataset used to train the model, and the capped distance during testing. Our algorithm significantly outperforms previous unsupervised algorithms using monocular video. Struct2depth [1] uses a pretrained model to initialize the network, which will improve the performance [30]. Besides, it uses a segmentation mask during training. Our algorithm learns from scratch without requiring extra information and outperforms struct2depth [1], except for the last 2 metrics. Furthermore, our algorithm yields comparable or better results than the algorithms that use stereo sequences to train the network (*i.e.*, pose is used during the training).

Fig. 6 provides some qualitative results on the KITTI dataset. Previous work provides a low resolution depth map, so their results are generally blurry and lose some details of the image. Furthermore, their algorithms may generate false depth estimation. For example, in the last column of Fig. 6, only our algorithm correctly recovers the wall. Besides, struct2depth [1] sometimes assigns the wrong depth to the sky. However, these errors are not counted, as they are out of the range of the LiDAR.

Make3D We also used the Make3D dataset [21] to test the cross dataset generalization ability of our algorithm. Our model is trained on the Cityscapes+Kitti dataset, then tested on the Make3D dataset. Table II lists the results. Our algorithm gives the state-of-the-art results.

Computational Complexity Our algorithm extends the U-Net and generates more constraints for training. This increases the inference and training time. The training takes about 35 hours on a single GTX 1080. The depth inference takes about 16 ms. It is slower than the original U-Net (9 ms). Although our depth net increases the computational complexity, the inference speed still achieves real-time.

C. Ablation Study

We conduct an ablation study to investigate the contributions from each component. Specifically, we add one component to the baseline or remove it from the full system. Table III lists the results. The baseline (first row of Table III) has a U-Net structure, and uses the cost similar to the temporal component in [14] except that we do not include the normal constraints in the cost function. Compared to the baseline, our WBAF doubles the camera baseline, and introduces more than 2 times the number of constraints. This significantly improves the accuracy. It is obvious that each of the introduced components contributes to an improvement in the performance. We find that adding WBAF, SRN or SPN alone can provide a comparable result to [12], [9], [7]. Removing one of the components reduces the accuracy, but this still provides better results than [12], [9], [7].

TABLE I

MONOCULAR DEPTH PREDICTION RESULTS ON THE KITTI DATASET USING THE SPLIT OF EIGEN. THE DATASET COLUMN LISTS THE TRAINING DATASET. **K** AND **CS** DENOTE THE KITTI AND CITYSCAPES DATASET, RESPECTIVELY. THE **SUP.** COLUMN DENOTES THE SUPERVISION LEVEL. **D**, **P** AND **SM** REPRESENT DEPTH, POSE AND SEGMENTATION MASK. THE RESULTS ARE EVALUATED FOR THE DEPTH CAPPED AT $80m$ AND $50m$. STRUCT2DEPTH [1] USES PRETRAINED RESNET18 MODEL. OTHERS LEARN FROM SCRATCH. THE BEST RESULTS OF EACH PART ARE IN BOLD.

Method	Sup.	Dataset	Cap	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen [22]	D	K	80	0.203	1.548	6.307	0.282	0.702	0.890	0.958
PyD-Net(200) [37]	P	K	80	0.153	1.363	6.030	0.252	0.789	0.918	0.963
Godard [6]	P	K	80	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Zhan [8]	P	K	80	0.135	1.132	5.58	0.229	0.820	0.933	0.971
Yang [31]	P	K	80	0.127	1.239	6.247	0.214	0.847	0.926	0.969
Zhan [14]	P	K	80	0.133	1.083	5.580	0.229	0.816	0.932	0.971
SuperDepth [19]	P	K	80	0.112	0.875	4.958	0.207	**0.852**	**0.947**	0.977
BA-Net [17]	P+D	K	80	**0.083**	**0.025**	**3.640**	**0.134**	-	-	-
Struct2depth [1]*	SM	K	80	0.141	1.026	5.291	0.215	0.816	0.945	**0.979**
Kumar [38]	No	K	80	0.211	1.980	6.154	0.264	0.732	0.898	0.959
Yang [10]	No	K	80	0.182	1.481	6.501	0.267	0.725	0.906	0.963
SfMLearner++ [32]	No	K	80	0.175	1.396	5.986	0.255	0.756	0.917	0.967
Mahjourian [13]	No	K	80	0.163	1.240	6.220	0.250	0.762	0.916	0.968
LEGO [39]	No	K	80	0.162	1.352	6.276	0.252	-	-	-
GeoNet [12]	No	K	80	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DDVO [9]	No	K	80	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net [7]	No	K	80	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Ours	No	K	80	**0.135**	**0.992**	**5.288**	**0.211**	**0.831**	**0.942**	**0.976**
Zhou [5]	No	K	50	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Mahjourian [13]	No	K	50	0.155	0.927	4.549	0.231	0.781	0.931	0.975
GeoNet [12]	No	K	50	0.147	0.936	4.348	0.218	0.810	0.941	0.977
Ours	No	K	50	**0.129**	**0.767**	**3.982**	**0.197**	**0.847**	**0.949**	**0.980**
Zhou [5]	No	CS+K	80	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian [13]	No	CS+K	80	0.159	1.231	5.912	0.243	0.784	0.923	0.970
LEGO [39]	No	CS+K	80	0.159	1.345	6.254	0.247	-	-	-
GeoNet [12]	No	CS+K	80	0.153	1.328	5.737	0.232	0.802	0.934	0.972
DDVO [9]	No	CS+K	80	0.148	1.187	5.496	0.226	0.812	0.938	0.975
Ours	No	CS+K	80	**0.135**	**1.022**	**5.188**	**0.211**	**0.835**	**0.943**	**0.977**

TABLE II

RESULTS ON THE MAKE3D DATASET. WE DIRECTLY APPLY THE MODEL TRAINED ON CS+K TO THE TEST DATASET OF MAKE3D. ERRORS ARE CALCULATED FOR PIXELS WHOSE DEPTHS ARE LESS THAN 70 METERS.

Method	Sup.	Abs Rel	Sq Rel	RMSE	RMSE log
Karsch [40]	Depth	0.428	5.079	8.389	0.149
Laina [41]	Depth	**0.204**	**1.840**	**5.683**	**0.084**
Godard [6]	Pose	0.544	10.94	11.76	0.193
Zhou [5]	No	0.383	5.321	10.47	0.478
DDVO [9]	No	0.387	4.720	8.09	0.204
Ours	No	**0.347**	**3.518**	**7.303**	**0.188**

TABLE III

ABLATION RESULTS. TO STUDY THE CONTRIBUTION OF INTRODUCED COMPONENTS, EACH OF THEM IS ADDED TO THE BASELINE, OR IS REMOVED FROM THE FULL SYSTEM. CL REPRESENTS THE CLIP LOSS.

WBAF	SPN	SRN	CL	Abs Rel	Sq Rel	RMSE	RMSE log
✖	✖	✖	✖	0.172	1.197	6.052	0.248
✔	✖	✖	✖	0.150	1.099	5.635	0.227
✖	✔	✖	✖	0.152	1.093	5.571	0.229
✖	✖	✔	✖	0.157	1.127	5.594	0.232
✖	✖	✖	✔	0.161	1.125	5.625	0.235
✔	✔	✔	✖	0.139	1.014	5.413	0.217
✔	✔	✖	✔	0.142	1.024	5.501	0.218
✔	✖	✔	✔	0.146	1.077	5.423	0.222
✖	✔	✔	✔	0.145	1.023	5.528	0.219
✔	✔	✔	✔	**0.135**	**0.992**	**5.288**	**0.211**

We also tested the impact of the parameter q of the clip loss function (6). The results in table IV show that the clip loss function could improve the performance for a large range of q in the KITTI dataset. However, how to determine an optimal q for each training sequence is still an open question. We will investigate this topic in our future work.

TABLE IV

IMPACT OF THE PERCENTILE OF THE CLIP LOSS FUNCTION

percentile	Abs Rel	Sq Rel	RMSE	RMSE log
100	0.139	1.014	5.413	0.217
98	0.137	1.006	5.388	0.216
95	0.135	0.992	5.288	0.211
92	0.136	0.990	5.296	0.213
90	0.138	1.020	5.401	0.218

D. Pose Estimation

We evaluate the performance of our pose estimation model using the KITTI odometry dataset The test sequences are split into 5-frame snippets, and the Absolute Trajectory Error (ATE) [5] is adopted as the metric. We adopt online refinement introduced in [1] . Table V lists the results. Our algorithm achieves the same result as [1] in sequence 9, and outperforms all of the competing algorithms in sequence 10.

IV. CONCLUSION

This paper presents a WBAF to train the network. Compared to previous work, our framework generates more constraints and a larger camera baseline. We apply a SPN and a SRN to extend the traditional U-Net. The SPN addresses the context information reduction problem of the U-

TABLE V

POSE ESTIMATION RESULTS ON THE KITTI DATASET [42].

	Seq. 09	Seq.10
ORB-SLAM (full)	0.014 ± 0.008	0.012 ± 0.011
ORB-SLAM (short)	0.064 ± 0.141	0.064 ± 0.130
Zhou *et al.* [5]	0.021 ± 0.017	0.020 ± 0.015
Mahjourian *et al.* [13]	0.013 ± 0.010	0.012 ± 0.011
DF-Net [7]	0.017 ± 0.007	0.015 ± 0.009
Yin *et al.* [12]	0.012 ± 0.007	0.012 ± 0.009
structure2depth [1]	$\mathbf{0.011 \pm 0.006}$	0.011 ± 0.010
Ours	$\mathbf{0.011 \pm 0.006}$	$\mathbf{0.010 \pm 0.009}$

Net. The SRN solves the problem caused by interpolation which may result in a blurry depth map and ignore details. Furthermore, we introduce a clip loss function that can make the training robust to moving objects and occlusions. Experimental results show that the presented components can benefit depth estimation, and our algorithm yields the state-of-the-art results.

REFERENCES

[1] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *AAAI*, 2019.

[2] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs," in *Proceedings of CVPR*, 2015, pp. 1119–1127.

[3] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *CVPR*, 2015, pp. 5162–5170.

[4] F. Liu, C. Shen, G. Lin, and I. D. Reid, "Learning depth from single monocular images using deep convolutional neural fields." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, 2016.

[5] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of CVPR*, vol. 2, no. 6, 2017, p. 7.

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, vol. 2, no. 6, 2017, p. 7.

[7] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proceedings of ECCV*. Springer, 2018, pp. 38–55.

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

[9] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proceedings of CVPR*, 2018, pp. 2022–2030.

[10] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry from videos with edge-aware depth-normal consistency," in *AAAI*, 2018, pp. 7493–7500.

[11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of CVPR*, vol. 2, 2018.

[13] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of CVPR*, 2018, pp. 5667–5675.

[14] H. Zhan, C. S. Weerasekera, R. Garg, and I. Reid, "Self-supervised learning for single view depth and surface normal estimation," in *ICRA*, May 2019, pp. 4811–4817.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[17] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," in *ICLR*, 2019.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[19] S. Pillai, R. Ambruş, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9250–9256.

[20] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[21] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[22] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proceedings of NIPS*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2366–2374.

[23] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, "Real-time joint semantic segmentation and depth estimation using asymmetric annotations," in *ICRA*. IEEE, 2019, pp. 7101–7107.

[24] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *CVPR*, 2018, pp. 2041–2050.

[25] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Proceedings of ECCV*. Springer, 2016, pp. 740–756.

[26] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Proceedings of NIPS*, 2015, pp. 2017–2025.

[27] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *ECCV Workshops*, 2018.

[28] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[29] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *Proceedings of ECCV*, 2018, pp. 484–500.

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

[31] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding," in *ECCV*. Springer, 2018, pp. 691–709.

[32] V. Prasad and B. Bhowmick, "Sfmlearner++: Learning monocular depth & ego-motion using meaningful geometric constraints," *arXiv preprint arXiv:1812.08370*, 2018.

[33] 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, 2018.

[34] 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.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of CVPR*, 2015, pp. 3061–3070.

[37] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *IROS*, 2018.

[38] A. CS Kumar, S. M. Bhandarkar, and M. Prasad, "Monocular depth prediction using generative adversarial networks," in *Proceedings of CVPR Workshops*, 2018, pp. 300–308.

[39] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Lego: Learning edge with geometry all at once by watching videos," in *Proceedings of CVPR*, 2018, pp. 225–234.

[40] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE transactions on PAMI*, vol. 36, no. 11, pp. 2144–2158, 2014.

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

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