View Synthesis with Multi-scale Cost Aggregation and Confidence Prior

Qi Wu
School of Computer Science
Northwestern Polytechnical University
Xi'an 710072, China
dongdawuqi@mail.nwpu.edu.cn

Xue Wang
School of Computer Science
Northwestern Polytechnical University
Xi'an 710072, China
xwang@nwpu.edu.cn

Qing Wang
School of Computer Science
Northwestern Polytechnical University
Xi'an 710072 and
Pazhou Lab, Guangzhou, China
qwang@nwpu.edu.cn

***Abstract*—This paper presents a learning-based novel view synthesis (NVS) approach from wide-baseline image pairs. Inspired by prior work, we first predict a depth probability volume which represents the scene structure as a set of depth probability layers (DPLs) within a reference view frustum. To reduce geometric uncertainty in ambiguous regions between input images, a multi-scale cost aggregation network is proposed to generate the DPLs for both input views without supervision. Furthermore, to mitigate the depth discretizaiton artifacts in distant views, we calculate the disparity map of the target view by passing the warped DPLs onto the target view to a CNN-based fusion network. Finally the predicted view could be obtained by incorporating the disparity map, warped input images and the confidence prior together. The proposed method improves the performance on challenging scenarios such as texture-less or non-textured regions, occlusion boundaries, non-Lambertian surfaces, and distant viewpoints. Experimental results show that our method achieves state-of-the-art view interpolation and extrapolation results on RealEstate10K mini dataset.**

***Index Terms*—view synthesis, sparse views, multi-scale cost aggregation, wide baseline, confidence prior**

Fig. 1. Given two images taken from distant viewpoints, our algorithm predicts a DPL-based scene representation that can render view interpolation and extrapolation. The left column: input image pair. The middle column: ground truth (top) and predicted disparity map (bottom). The right column: predicted target views by MPI [1] (top) and our method (bottom).

I. Introduction

NVS is a classic problem in vision and graphics, from which one can generate any view within an appropriate region given a set of input views. Rendering novel views could offer computational photographic results with various visual effects, such as defocus, 3D movies, 3D Ken Burns and so on. Meanwhile, view synthesis techniques can be used to generate realistic free view videos for virtual reality and augmented reality. Image-based rendering (IBR) with sparse input images has made remarkable progress in past several years due to the emergence of deep learning and the popularity of portable photographic equipments.

There are several major issues in NVS. First, the ambiguity problem poses serious challenges for scene structure estimation as the baseline expands. For repeated texture or texture-less regions, it is difficult to estimate correspondence solely using appearance similarity. Texture tearing easily occurs under such circumstances. Appearance inconsistency is another more serious problem usually caused by varying illumination and occlusion across views. Such inconsistency will lead to severe artifacts at non-Lambertian and occlusion boundary areas. Second, unobserved regions in the target view. Generally, imprecise sampling or disocclusions will result in holes in the synthesized view. A post-processing optimization is generally needed to fill the holes.

In this paper, we propose a novel learning-based framework for NVS given an image pair. As shown in Fig. 1, our method outperforms prior work in two specific ways: (1) geometric uncertainty in ambiguous regions are mitigated, such as texture-less or non-textured regions, occlusion boundaries, and non-Lambertian surfaces, (2) noticeable visual artifacts produced by predicting dis-occluded scene contents in previous methods are alleviated (e.g. the gray sofa at the bottom-left corner). To reduce geometric uncertainty in ambiguous regions between input images, we design a multi-scale cost aggregation network to estimate the DPLs for both input views. Then, given camera parameters, we feed the warped DPLs of input views onto the target view to a CNN to generate the DPLs of the target view. The reconstructed DPLs are used to calculate a disparity map to reduce the depth discretizaiton artifacts in distant views. By combining the disparity map, warped input images and a confidence prior, the fusion network further learns to generate the synthesized novel view with realistic

The work was supported by NSFC under Grant 61801396 and 62031023.

2021 Digital Image Computing: Techniques and Applications (DICTA) | 978-1-6654-1709-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/DICTA52665.2021.9647048

visual effects.

In summary, the main contributions of this paper include:

(1) A novel DPL-based view synthesis solution is proposed, which achieves high quality view interpolation and extrapolation results even on challenging scenarios.

(2) The multi-scale cost aggregation network is designed for estimating dense and structure-maintained depths without supervision, which mitigates geometric uncertainty in ambiguous regions.

(3) A confidence prior is deployed to provide guidance for the fusion network to predict dis-occluded contents and correct inaccurate estimation in the target view.

II. Related Work

There is a large literature on NVS and image-based rendering. In this section, we discuss the most relevant research works to our method.

A. Traditional Methods

Early methods [2], [3] directly study the combination of input images to generate novel views when the structure information is available. Clearly, structure estimation is an inseparable part of view synthesis. However it is difficult to build pervasive mixture models for complex scenes. For searching more accurate geometric correspondences, Chauraisa *et al.* [4] estimate scene depths based on over-segmentation and then project the super pixels to the target view to perform multi-view fusion. However, the spatial discreteness from over-segmentation may cause semantic tearing or structural distortion. By defining scenario goals, Hoiem *et al.* [5] propose a photo pop-up method using a sparse structure model. Woodford *et al.* [6] exploit multi-labels conditional random fields to estimate both depths and colors of novel views.

Many view synthesis methods [4], [7]–[9] follow the framework where the geometry structure for each input is estimated first and the novel view is then rendered by multi-view fusion. However, it is still difficult to build pervasive models to handle the ambiguity in complex scenes with traditional methods.

B. Learning-based Methods

Recent work has demonstrated the effectiveness of deep learning for NVS. Flynn *et al.* [10] treat the NVS task as the blending of multiple color layers and selection volumes. Zhou *et al.* [11] directly learn the appearance flow between input and output views. Lacking reasoning about the scene structure, these methods may cause structure distortion and detail blur. To learn a more accurate depth map, Kalantari *et al.* [12] use two convolutional neural networks (CNNs) to estimate depths and pixel colors of the target image respectively. They apply the shifting operation on the source images at each disparity level to estimate pixel correspondences which leads to more robust estimation of target depths. However, the correspondences are weakened by network convolution and become unreliable as the baseline expands. Although another CNN is adopted for error correction and occlusion areas, the correction itself is an ill-posed problem.

To handle this problem, many researchers [1], [13]–[16] propose the layer-based representation, i.e. a distribution over depth, for the scene structure, which is capable of representing geometric uncertainty in ambiguous regions [15]. Tulsiani *et al.* [14] attempt to learn layered depth images (LDIs) for the scene. However, under the supervision of view synthesis, it is difficult to predict LDIs without explicit corresponding relationship. Zhou *et al.* [1] predict a Multi-Planar Image (MPI) composed of RGB and alpha layers from two input images directly via a learned feed-forward network. The method passes the input images to the network as a plane sweep volume (PSV) which removes the need to explicitly supply the camera pose, and also allows the network to more efficiently determine correspondences between images. Later, NVS are extended to the extrapolation regime based on MPI [15], [17]. However, such networks have no intrinsic ability to understand visibility between input views and the predicted MPI, instead they rely on the network layers to learn geometry.

To overcome this, Flynn *et al.* [16] consider the differentiable inverse problem of generating MPI and present a gradient-learning method. Tucker *et al.* [18] further apply MPI to the single-view synthesis problem. Given extra point clouds for normalization and supervision, discrete MPIs are fused to generate a disparity map. Choi *et al.* [13] predict depth probability volumes of the target view with an image refinement network [19]. They obtain a robust estimate of scene structure from cost volumes in high dimensions, which provides more robust correspondences but ignores low level details. Shi *et al.* [20] exploit multi-scale VGG features to restore high frequency details after depth image based rendering (DIBR), which depends on accurate depth estimation.

Instead of solely depending on single-scale features, we aggregate multi-scale explicit geometric correspondences for more robust scene structure estimation. To our best knowledge, Xu *et al.* [21] adopts a similar multi-scale aggregation idea for disparity estimation. However, their method is proposed to remove the 3D convolution and limited to stereo matching with supervision, which significantly differs from our method.

III. Methodology

A. Overview

Generally, given two input images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ taken from two cameras C_1 and C_2 respectively, our goal is rendering arbitrary novel views, including view interpolation and extrapolation within a certain range. Let $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ represent the intrinsic camera parameters of C_1 and C_2 respectively. By regarding the camera coordinate of C_1 as the world coordinate, the extrinsic parameters comprising a rotation matrix and a translation vector for C_2 is $[\boldsymbol{R}\ \boldsymbol{t}]$. In this work, we consider a simplified case that the input image pair is taken from one dynamically moving camera, therefore their intrinsic camera parameters are the same, which can be referred as $\boldsymbol{K}$.

Fig. 2 depicts the proposed framework. Firstly, the DPLs for both $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are estimated by a well-designed multi-scale cost aggregation network. Then the DPLs of the target view can be obtained by fusing warped DPLs from each input view

using a full convolution network, in which layered confidence priors are adopted to guide this process. Moreover, we choose the fusion way of [18] to obtain a disparity map at the target location, by which $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are backward projected to the target view. Finally, a confidence prior volume is calculated to blend warped synthesized views for further optimization.

B. Depth Estimation

1) Definition of DPLs: Inspired by [13], we exploit the feasibility of using layered depth probability to estimate the scene structure. The major difference with [13] is that the our layers are essentially taken from the alpha layers of MPI [1], and the scene could be deconstructed using less such layers. The DPL volume consists of a stack of depth probability layers $\boldsymbol{p}_l \in \mathbb{R}^{H\times W}$ uniformly sampled according to disparity (inversely proportional to depth):

$$\mathrm{DPLs} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_D\}, \tag{1}$$

where D denotes the number of layers. Note that, the smaller the subscript of a layer, the farther the layer to the image plane.

By regarding the image pixel as a light ray emitting from the corresponding scene point, we can calculate a soft probability distribution along the depth direction, i.e. the z-axis of the camera coordinate. The depth value with the highest probability indicates the real depth of the scene point. In stereo matching, this probability is inversely proportional to the cost volume [22]. Here we formate the DPL prediction problem as a process of multi-scale cost calculation and aggregation.

2) Multi-scale cost aggregation: It is well known that repeated textures, occlusion boundaries, non-Lambertian surfaces tend to deteriorate correctly matched similarity measures and introduce more ambiguity as the baseline increases (Fig. 3). To address this problem, we propose a robust multi-scale cost aggregation network for estimating DPLs. For an input image, we extract two feature levels from a feature pyramid network (FeaEx) composed of two downsampling layers and the Atrous Spatial Pyramid Pooling (ASPP) module similar to DeepLab V3 [23]:

$$\left\{\boldsymbol{F}_i^2, \boldsymbol{F}_i^3\right\} = \mathrm{FeaEx}(\boldsymbol{I}_i), i = 1, 2. \tag{2}$$

where $\boldsymbol{F}_i^2 \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}}$, $\boldsymbol{F}_i^3 \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}}$ are the feature maps with different resolutions, which represent the characteristic space on two different scales.

To estimate depth probability volumes combing $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, for a certain scale s, we apply the backward projection operation on the feature maps of $\boldsymbol{I}_2$ (BackPro) to get the re-projected feature volume $\{\tilde{\boldsymbol{F}}_{2\to1}^s\}$ by stacking all the feature maps over the feature and depth dimensions together:

$$\left\{\tilde{\boldsymbol{F}}_{2\to1}^s\right\} = \mathrm{Stack}\left\{\mathrm{BackPro}(\boldsymbol{F}_2^s, \boldsymbol{K}^s, \boldsymbol{R}, \boldsymbol{t}, d_l)_{l=1:D_s}\right\}, s = 2, 3, \tag{3}$$

where $\boldsymbol{K}^s$ is the intrinsic matrix at scale s, d_l denotes the depth value of layer l. The number of depth layers D_s at scale s meets $D_s = \dfrac{D_3}{2^{(s-3)}}$. In this way, the feature volume $\{\tilde{\boldsymbol{F}}_{2\to1}^s\}$ takes the dimensions $[\dfrac{H}{2^{(s-1)}}, \dfrac{W}{2^{(s-1)}}, \dfrac{D_3}{2^{(s-3)}}, \dfrac{C}{2^{(s-1)}}]$, where C denotes the number of feature maps in the first convolutional layer. To enable identical dimensions with $\{\tilde{\boldsymbol{F}}_{2\to1}^s\}$, we duplicate $\boldsymbol{F}_1^s$ to $\{\tilde{\boldsymbol{F}}_1^s\}$ at different disparities. Instead of directly subtracting two volumes, we adopt the concatenation operation to aggregate corresponding features across scale using

$$\boldsymbol{C}_1^s = \mathrm{Concat}(\tilde{\boldsymbol{F}}_{2\to1}^s, \tilde{\boldsymbol{F}}_1^s), s = 2, 3. \tag{4}$$

For more accurate and smoother DPLs, we exploit a 3D CNN and a channel attention module to aggregate contextual information for different dimensions and simultaneously learn the weights among the feature dimension as show in Fig. 2. This module is abbreviated as CA. We aggregate the cost volume $\tilde{\boldsymbol{C}}_1^s$ at scale s by

$$\tilde{\boldsymbol{C}}_1^s = \mathrm{CA}(\boldsymbol{C}_1^s), s = 2, 3. \tag{5}$$

Furthermore, upsampled $\tilde{\boldsymbol{C}}_1^3$ and $\tilde{\boldsymbol{C}}_1^2$ are concatenated as the input to another CA module and the DPLs of $\boldsymbol{I}_1$ could be obtained by:

$$\boldsymbol{C}_1^{fin} = \mathrm{CA}\left(\mathrm{Concat}(\tilde{\boldsymbol{C}}_1^2, \tilde{\boldsymbol{C}}_1^3 \uparrow)\right), \tag{6}$$

$$\left\{\boldsymbol{p}_1^1, ..., \boldsymbol{p}_{D_1}^1\right\} = \boldsymbol{C}_1^{fin} \uparrow. \tag{7}$$

Due to the inverse relation between cost and probability, after applying the activation function $tanh$, DPLs will take negative values of the original results. Similarly, we can compute the DPLs of $\boldsymbol{I}_2$ by exchanging input pairs following the above process $\boldsymbol{f}$:

$$\left\{\boldsymbol{p}_1^2, ..., \boldsymbol{p}_{D_1}^2\right\} = \boldsymbol{f}\left(\boldsymbol{I}_2, \boldsymbol{I}_1\right). \tag{8}$$

3) Warping of DPL: To render a novel view $\boldsymbol{I}'$, of which the intrinsic and extrinsic camera parameters are represented by $\boldsymbol{K}'$ and $[\boldsymbol{R}'\ \boldsymbol{t}']$ respectively, each layer in the DPLs of the input view is warped to the synthesized view as follows:

$$\tilde{\boldsymbol{p}}_l^1 = \mathrm{Warp}\left(\boldsymbol{p}_l^1, d_l\right),\ \tilde{\boldsymbol{p}}_l^2 = \mathrm{Warp}\left(\boldsymbol{p}_l^2, d_l\right),\ l = 1, ..., D_1. \tag{9}$$

We adopt the homography-based inverse warping on each depth layer. The target pixel (u_t, v_t) is estimated by bilinearly sampling from the neighbors of the source pixel (u_s, v_s) computed as:

$$\begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} \sim \boldsymbol{K}_s\left(\boldsymbol{R}_t - \frac{\boldsymbol{t}_t\boldsymbol{n}^\top}{d_l}\right)\boldsymbol{K}_t^{-1}\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix}, \tag{10}$$

where $\boldsymbol{K}_s$ and $\boldsymbol{K}_t$ are the source and target camera intrinsics, $\boldsymbol{R}_t$ and $\boldsymbol{t}_t$ are the rotation matrix and translation vector of the target view with respect to the source view respectively. The vector $\boldsymbol{n} = (0\ 0\ 1)^\top$ is the normal vector of the depth plane.

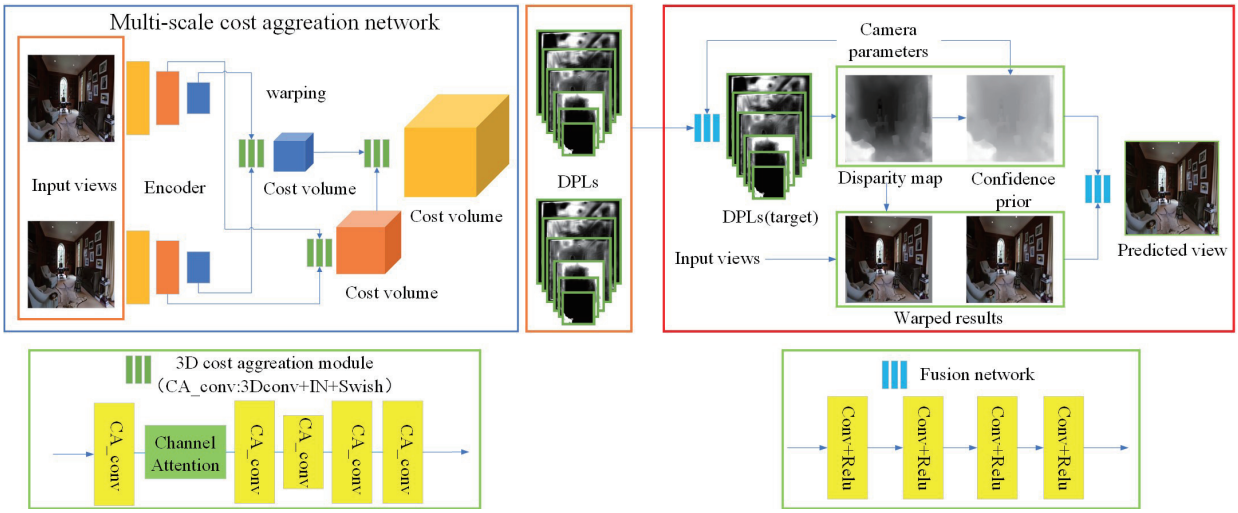

Fig. 2. Overview of the proposed end-to-end framework for synthesizing a novel view within or beyond the baseline given a stereo input. The multi-scale cost aggregation network is used to estimate the DPLs for each input. The DPLs are then projected to the target view and further blended through the fusion network to generate the disparity map of the target view. Finally, warped results using the disparity map are merged to obtain the final predicted view under the guidance of the prior information.

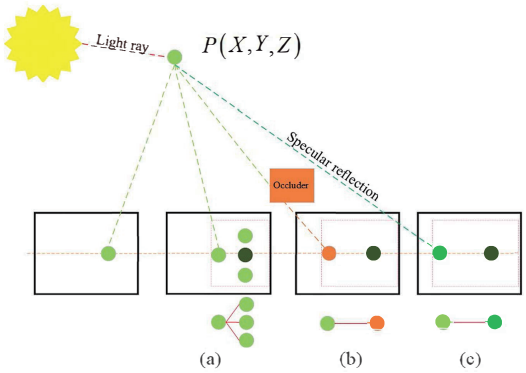

Fig. 3. Challenging situations for correspondence estimation as baseline increases. (a) Ambiguity caused by repeated and textless texture. (b) Occlusions. (c) Specular reflection of non-Lambertian surfaces. The dark green point is the projection of the green point in the left view. The rectangle marked in dotted lines represents searching area.

4) Fusion and composition: A learning-based fusion module (Fus) is proposed to merge $\tilde{\boldsymbol{p}}_l^1$ and $\tilde{\boldsymbol{p}}_l^2$. To make the fusion results more reliable, we calculate a fusion prior m_l for each layer by combining camera parameters and the Mean Squared Error (MSE) value of $\boldsymbol{t}'$, t_{loss}, as follows:

$$m_l = e^{-\frac{t_{loss} f_x}{d_l}}, l = 1, ..., D_1, \tag{11}$$

where f_x is the focal length. t_{loss} measures the position relation in the physical space, which is converted to the pixel space by Eq. 11. Then, a fusion mask volume $\boldsymbol{M} \in \mathbb{R}^{H \times W \times D_1}$ is learned by the fusion module using

$$\boldsymbol{M} = \mathrm{Fus}\left(\left\{\tilde{\boldsymbol{p}}_l^1, \tilde{\boldsymbol{p}}_l^2, m_l\right\}_{l=1:D_1}\right). \tag{12}$$

After obtaining the fusion mask volume $\boldsymbol{M}$, the DPLs of $\boldsymbol{I}'$ is calculated by

$$\left\{\boldsymbol{p}'_1, ..., \boldsymbol{p}'_{D_1}\right\} = \boldsymbol{M} \otimes \left\{\tilde{\boldsymbol{p}}_1^1, ..., \tilde{\boldsymbol{p}}_{D_1}^1\right\} + (\mathbf{1} - \boldsymbol{M}) \otimes \left\{\tilde{\boldsymbol{p}}_1^2, ..., \tilde{\boldsymbol{p}}_{D_1}^2\right\}, \tag{13}$$

where $\otimes$ denotes the dot production operation. Based on the DPLs of $\boldsymbol{I}'$, the disparity map $\boldsymbol{Disp}$ of the synthesized view can be computed using the over operation [1], [18], [24]:

$$\boldsymbol{Disp} = \sum_{l=1}^{D_1} \left(d_l^{-1} \boldsymbol{p}'_l \otimes \prod_{k=l+1}^{D_1} (\mathbf{1} - \boldsymbol{p}'_k) \right). \tag{14}$$

C. View Reconstruction

Given the disparity map $\boldsymbol{Disp}$, we can backward project the target pixel (u', v') in $\boldsymbol{I}'$ to the input image to find the corresponding pixel. Taking $\boldsymbol{I}_1$ for an example, the relation is computed as follows:

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \sim \boldsymbol{K} \left(\boldsymbol{R}' \boldsymbol{K}'^{-1} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} + \boldsymbol{t}' \boldsymbol{Disp}(u', v') \right), \tag{15}$$

where (u_1, v_1) denotes the computed corresponding pixel in $\boldsymbol{I}_1$, and $\boldsymbol{Disp}(u', v')$ is the disparity value of (u', v').

The reconstructed target images $\boldsymbol{I}'_1$ and $\boldsymbol{I}'_2$, which are generated from $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ respectively, are still pretty rough due to inaccurate depths and disoccluded areas. Therefore, we adopt the same network structure in Eq. 12 to directly predict the reconstructed image $\tilde{\boldsymbol{I}}'$ as described in [12]. Moreover, we use the confidence prior matrix $\boldsymbol{C}_p \in \mathbb{R}^{H \times W}$ calculated using the camera parameters and disparities similar to Eq. 11, in which the disparity map $\boldsymbol{Disp}$ is used for correcting errors caused by inaccurate depths or disocclusions:

$$\boldsymbol{C}_p = e^{-t_{loss} f_x \boldsymbol{Disp}}, \tag{16}$$

$$\tilde{\boldsymbol{I}}' = \mathrm{Fus}\left(\boldsymbol{I}_1', \boldsymbol{I}_2', \boldsymbol{C}_p, \boldsymbol{Disp}\right). \tag{17}$$

D. Losses

Our total loss is composed of a view synthesis loss and a smooth loss [18]:

$$\mathcal{L} = \mathcal{L}^{vgg} + \mathcal{L}^{smooth}. \tag{18}$$

Many previous similar works [1], [15], [20] have shown that the perceptual loss helps improve the performance over the pixel-level reconstruction loss and the same is true for our work. Therefore, we use the perceptual loss for layers from VGG-19 [25], where the implementation is similar to Zhou *et al.* [26]. Moreover, there exists a variety of losses for smooth depth preiction [27]–[29]. However, these smooth losses may be unconvincing without the guidance. On the contrary, we choose the smooth loss described in [18], which could smooth small gradient regions and enable large gradient regions aligned to image edges:

$$\boldsymbol{E} = \min\left(\frac{G(\boldsymbol{I}')}{e_{min} \times \max_{(x,y)} G(\boldsymbol{I}')}, \mathbf{1}\right), \tag{19}$$

where the operation $G(\cdot)$ is the gradient sum across all channels and e_{min} is the fraction of the maximum gradient. $\boldsymbol{E}$ is actually a source edge mask used to distinguish between large gradient and small gradient regions in the target image.

The smooth loss for the target view can be defined as follows by combing the source edge mask $\boldsymbol{E}$,

$$\mathcal{L}_t^{smooth} = \frac{1}{N} \sum_{(x,y)} \left(\max\left(G(\boldsymbol{Disp}) - g_{min}, 0\right) \odot \left(\mathbf{1} - \boldsymbol{E}\right)\right), \tag{20}$$

where $\odot$ is the Hadamard product and g_{min} is the threshold for gradient. The above loss could penalize large gradient areas in the disparity map for small gradients in the target image. Similar to [18], here we set $e_{min} = 0.1$ and $g_{min} = 0.05$.

When calculating the target DPLs, the smooth loss may effect the consistency of DPLs between the source view and the target view. Therefore, we generate the disparity map $\boldsymbol{Disp}_1$ for $\boldsymbol{I}_1$ as well according to Eq. 14 and smooth it in the same way. Our final smooth loss can be computed as:

$$\mathcal{L}^{smooth} = \alpha \mathcal{L}_t^{smooth} + \beta \mathcal{L}_1^{smooth}, \tag{21}$$

where α and β are weight coefficients.

The pseudocode of our proposed view synthesis is described in Algorithm 1.

Algorithm 1 DPL-based Free View Synthesis

Input: Images pair $\boldsymbol{I}_1, \boldsymbol{I}_2$ with known camera parameters $\boldsymbol{K}$, $\boldsymbol{R}$, $\boldsymbol{t}$, the camera parameters of the synthesized view $\boldsymbol{K}'$, $\boldsymbol{R}'$, and $\boldsymbol{t}'$

Output: A novel image $\tilde{\boldsymbol{I}}'$ at the target location

1: **for** each input image $\boldsymbol{I}_i, i = 1, 2$ **do**
2: generate $\left\{\boldsymbol{p}_1^i, ..., \boldsymbol{p}_{D_1}^i\right\}$ according to Eq.1-8
3: warp $\left\{\boldsymbol{p}_1^i, ..., \boldsymbol{p}_{D_1}^i\right\}$ to $\left\{\tilde{\boldsymbol{p}}_1^i, ..., \tilde{\boldsymbol{p}}_{D_1}^i\right\}$ of the target view according to Eq.9-10
4: **end for**
5: calculate the fusion mask volume $\boldsymbol{M}$ by Eq.11-12
6: merge $\left\{\tilde{\boldsymbol{p}}_1^i, ..., \tilde{\boldsymbol{p}}_{D_1}^i\right\}$ to get $\left\{\boldsymbol{p}_1', ..., \boldsymbol{p}_{D_1}'\right\}$ using $\boldsymbol{M}$ according to Eq.13
7: calculate the disparity map $\boldsymbol{Disp}$ for the target view refer to Eq.14
8: **for** each input image $\boldsymbol{I}_i$ **do**
9: obtain $\boldsymbol{I}_i'$ by backward warping $\boldsymbol{I}_i$ with $\boldsymbol{Disp}$ refer to Eq.15
10: **end for**
11: calculate the confidence prior C_p by Eq.16
12: merge $\boldsymbol{I}_1'$ and $\boldsymbol{I}_2'$ guided by C_p and $\boldsymbol{Disp}$ to predict $\tilde{\boldsymbol{I}}'$ refer to Eq.17
13: **return** $\tilde{\boldsymbol{I}}'$

IV. Experimental Results

The following section presents quantitative and qualitative evidences to validate the benefits of our method. We adopt two well-known objective image quality metrics, the peak-signal-to-noise ration (PSNR) as well as the structural similarity index measure (SSIM) for quantitative analysis. For both PSNR and SSIM, a higher value indicates that the image is of higher quality and vice-versa.

A. Dataset

We train and evaluate our method on the dataset extracted from the Real Estate 10K dataset [1] called Real Estate Mini (REM) dataset, consisting of 2549 sequences for training and 85 sequences for testing with known camera pose for each video frame. Compared to other datasets, REM involves more complex and varied scenarios, such as indoor, outdoor, mirror objects, fine structured object and so on. We generate training triplets following the method described in [1] by randomly sampling two source frames and a target frame from a randomly chosen video. The target view could be located within or beyond the baseline, which demonstrates the capacity of the proposed method for both view interpolation and extrapolation. The results are averaged over randomly sampled 1000 test triplets from the whole testing sequences.

B. Training Details

We specify the minimal number of depth layers $D_3 = 16$ for the nearest depth plane $d_{16} = 1\mathrm{m}$ to the farthest plane $d_1 = 100\mathrm{m}$. The weight coefficients α and β in Eq. 21 are set to 0.5 and 0.2 respectively. During training, the spatial resolution of input images is 192×192, however our model could be applied to arbitrary resolutions due to the full convolutional network. In order to improve the convergence speed and accuracy of our model, we initialize the network parameters using Xavier method [30] and adopt the ADAM optimizer [31] with the learning rate 0.00001, $\beta_1 = 0.9, \beta_2 = 0.999$. Due to the equipment memory limit, we set the batch size to be 1. The whole training process takes about 24 hours on 6 GTX 1080ti GPUs with the Tensorflow framework.

TABLE I
QUANTITATIVE COMPARISON RESULTS FOR DIFFERENT SAMPLING SETTINGS.

Metric	Method	5(in)	10(in)	5(ex)	10(ex)	5(ori)	10(ori)
PSNR	MPI [1]	26.57	22.87	25.41	22.70	31.98	26.28
	Ours	**30.29**	**26.01**	**28.01**	**23.66**	**34.43**	**29.53**
SSIM	MPI [1]	0.902	0.809	0.889	**0.812**	0.929	0.864
	Ours	**0.929**	**0.859**	**0.905**	0.806	**0.953**	**0.901**

C. Comparison to SOTA

We conduct a comparative analysis with one state-of-the-art algorithm MPI [1]. The superiority of MPI with regards to other existing methods is presented in [1].

We design six sampling patterns to verify the effectiveness of our method. The first two sampling settings are interpolation within 5 frames and 10 frames respectively, in which the frames on both ends are regarded as inputs and the other one as the target. The middle two settings are the extrapolation within 5/10 frames, in which the first two adjacent frames are inputs and the last one as the target. The final two settings randomly choose two input frames and one target frame among a triplet, similar to [1], which includes interpolation and extrapolation within 5/10 frames. These settings are referred as "5(in)", "10(in)", "5(ex)","10(ex)", "5(ori)" and "10(ori)" respectively.

For fairness, we retrain a 64-layer MPI model using the perception loss to enable its convergence. For the MPI method, the quality of the synthesized view is heavily dependent on the baseline width since the local pixel-correspondences are provided only by PSVs. Thus the ambiguity caused by repeated textures, occlusion boundaries, and non-Lambertian surfaces couldn't be handled well. Moreover, wilder baseline will further deteriorate its performance. In addition, the layers of uncertainty increase sampling for invalid areas, which is particularly serious with wide baseline inputs. On the contrary, geometric correspondences between different feature levels are aggregated by our method, which could provide high quality view synthesis results even with wide baseline inputs.

Table I shows quantitative comparisons between our method and MPI. The results indicate that our method generates more accurate synthesized views for both view interpolation and extrapolation, especially for view interpolation with wide-baseline inputs. However,the performance of our method degrades to a certain on 5(ex). Generally, the larger difference in the content between the target view and the input views, the more image regions not observed in both input views need to be estimated from the neighborhoods. This results in repeated textures in disoccluded areas from backward projection.

Qualitatively evaluations are shown in Fig. 4. Four different scenarios are included. (a) Scenes containing fine structures and repeated textures. The results of MPI show heavy artifacts on the window frame and doorsteps while our result shows fidelity in these areas. (b) Scenes containing varying occlusion relations. The results of MPI shows repeated textures at the boundaries of the pillar, sofa and treadmill. Our methods could generate clear and sharp boundaries in these areas. (c) Scenes containing non-Lambertian areas and repeated textures. The results of MPI produce aliasing effects in the black single sofa stool and the pillow areas with repeated texture, which introduces ambiguity in the areas of large disparities. Our method could handle this problem and generate more realistic results. (d) Outdoor scenes with more complicated structures and severe occlusions. Our method still outperforms MPI.

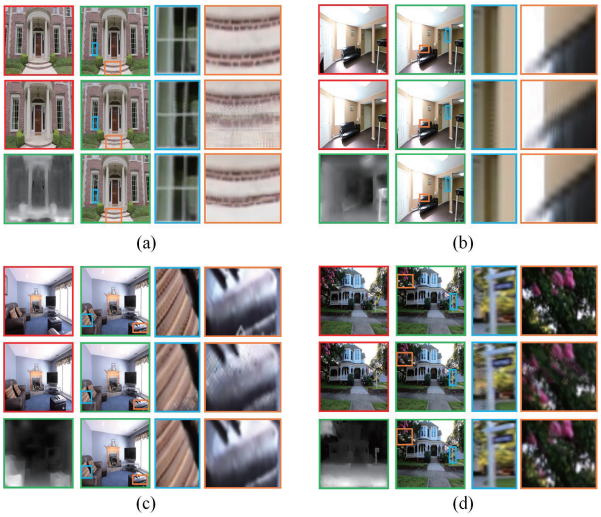

Fig. 4. Qualitative comparison of rendered novel views. Different scenarios are displayed from (a) to (d). For each subfigure, from top to bottom: input pairs and estimated disparity map of the target view (the leftmost column); ground truth, synthesized results of MPI and our method (the middle left column); enlarged local regions from ground truth, synthesized results of MPI and our method for more details (the middle right and rightmost columns).

Given the ground truth, we can compute the L1 error map for the reconstructed view. Fig. 5 demonstrates the comparisons between our method and MPI using L1 error maps. Similar results validate the efficiency and robustness of the proposed method for difficult scenes such as repeat textures, occlusion boundaries and non-Lambertian areas.

D. Ablation Study

To prove the effectiveness of our method, we conduct an ablation study. Three different strategies are considered: 1) traditional cost aggregation, 2) without cost aggregation, and 3) without the confidence prior.

Traditional cost aggregation methods solely exploit the aggregation of high-level features for cost volumes, while ignoring low level details. Besides it is difficult for these methods to convergence without depth supervision. For the compared method without cost aggregation, we adopt a framework similar to Unet, which includes an encoder and a decoder. Although the Unet framework is capable of restoring clear boundaries, it can't reduce the ambiguity or uncertainty arising from wide baselines without considering geometric correspondence. Table II shows quantitative analysis results. The complete model of the proposed method achieves the best performance. The sampling setting is set to 10(ori).

Partial results are shown in Fig. 6. Since there exists serious ambiguity in large disparity areas such as door edges, the sofa

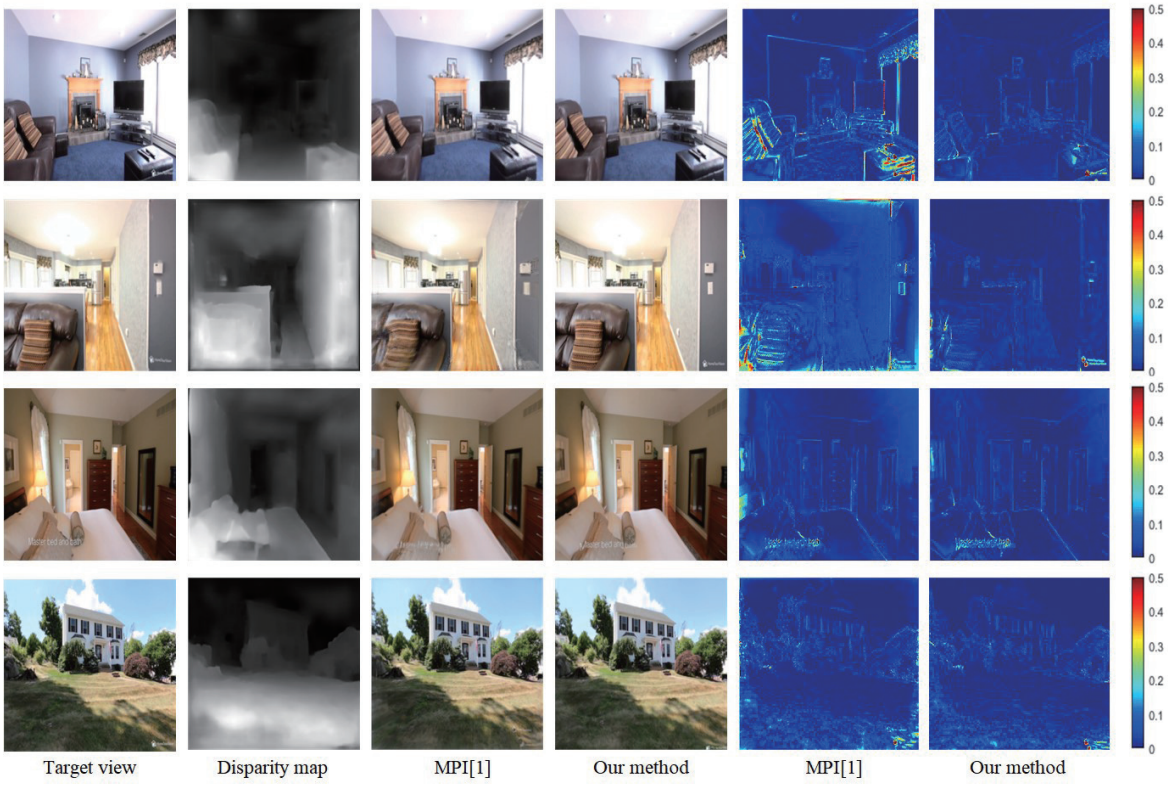

Fig. 5. Visual comparisons of rendered novel views with reconstruction error maps. From left to right: ground truth, estimated disparity map of the target view, synthesized results of MPI and our method, reconstruction error maps of MPI and our method.

TABLE II
QUANTITATIVE COMPARISON RESULTS FOR ABLATION STUDY.

Methods	PSNR	SSIM
traditional cost aggreation	26.05	0.793
w/o cost aggreation	26.81	0.802
w/o confidence prior	27.00	0.868
complete model	**29.53**	**0.901**

with weak textures, and reflective photo frames, the incomplete model without multi-scale cost aggregation tends to estimate inaccurate depths. Moreover, ghost effects at occlusion boundaries occur in the absence of the prior guidance.

V. CONCLUSIONS AND FUTURE WORK

We propose a learning-based framework for view synthesis from arbitrary sparse views with overlapping field of view (FOV). By regarding depth estimation as a multi-scale cost aggregation problem, the method could effectively reduce the ambiguity or uncertainty arising from repeat textures, occlusion boundaries and non-Lambertian surfaces and achieve high-quality NVS results for both view interpolation and extrapolation even in wide-baseline scenarios.

Currently we use a small amount of multiple scale layers and sampled layers. Also backward projection tends to introduce effects for wild-baseline view extrapolation. More efficient depth probability volumes and sophisticated warping techniques could further improve the performance.

REFERENCES

[1] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," in *SIGGRAPH*, 2018.

[2] S. Chen and L. Williams, "View interpolation for image synthesis," *Computer Graphics (SIGGRAPH'93)*, vol. 27, 09 2002.

[3] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 11–20. [Online]. Available: https://doi.org/10.1145/237170.237191

[4] G. Chaurasia, S. Duchene, O. Sorkinehornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Transactions on Graphics*, vol. 32, no. 3, p. 30, 2013.

[5] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, p. 577–584, Jul. 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073232

[6] O. Woodford, I. D. Reid, P. H. S. Torr, and A. W. Fitzgibbon, "On new view synthesis using multiview stereo," in *Proceedings of the 18th British Machine Vision Conference, Warwick*, vol. 2, 2007, pp. 1120–1129.

[7] E. S. Penner and L. Zhang, "Soft 3d reconstruction for view synthesis," *ACM Transactions on Graphics*, vol. 36, no. 6, p. 235, 2017.

[8] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, p. 600–608, Aug. 2004. [Online]. Available: https://doi.org/10.1145/1015706.1015766

[9] P. Hedman, J. Philip, T. Price, J. Frahm, G. Drettakis, and G. J. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Transactions on Graphics*, vol. 37, no. 6, p. 257, 2019.

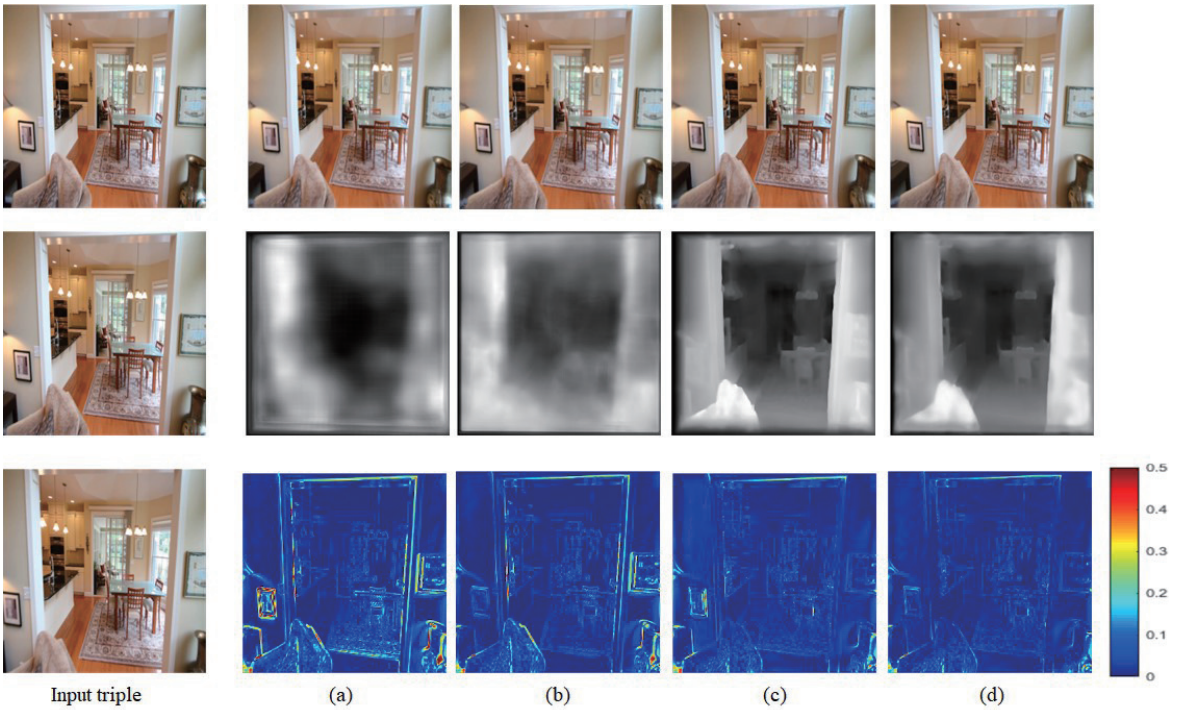

Fig. 6. Qualitative comparisons for ablation study. The leftmost column is the target view and two input views from top to bottom. Other columns show the results of different versions of our method: (a) Without multi-scale, (b) Without cost aggregation, (c) Without confidence prior, and (d) Complete model. From top to bottom: rendered target view, disparity map and L1 loss error map.

[10] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deep stereo: Learning to predict new views from the world's imagery," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5515–5524.

[11] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," *European Conference on Computer Vision (ECCV)*, 2016.

[12] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, vol. 35, no. 6, 2016.

[13] I. Choi, O. Gallo, A. Troccoli, M. H. Kim, and J. Kautz, "Extreme view synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7781–7790.

[14] S. Tulsiani, R. Tucker, and N. Snavely, "Layer-structured 3d scene inference via view synthesis." *European Conference on Computer Vision (ECCV)*, 2018.

[15] P. Srinivasan, R. Tucker, J. Barron, R. Ramamoorthi, R. Ng, and N. Snavely, "Pushing the boundaries of view extrapolation with multiplane images," in *IEEE Conference on Computer Vision and Pattern Recogni- tion (CVPR)*, 06 2019, pp. 175–184.

[16] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker, "Deepview: View synthesis with learned gradient descent," in *IEEE Conference on Computer Vision and Pattern Recogni- tion (CVPR)*, 06 2019, pp. 2362–2371.

[17] B. Mildenhall, P. P. Srinivasan, R. Ortizcayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics*, vol. 38, no. 4, p. 29, 2019.

[18] R. Tucker and N. Snavely, "Single-view view synthesis with multiplane images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[19] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] J. Shi, X. Jiang, and C. Guillemot, "Learning Fused Pixel and Feature-based View Reconstructions for Light Fields," in *CVPR 2020 - IEEE Conference on Computer Vision and Pattern Recognition*. Seattle, United States: IEEE, Jun. 2020, pp. 1–10. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02507722

[21] H. Xu and J. Zhang, "Aanet: Adaptive aggregation network for efficient stereo matching," *Computer Vision and Pattern Recognition*, 04 2020.

[22] H. 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)*, vol. 2, 2005, pp. 807–814 vol. 2.

[23] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv: Computer Vision and Pattern Recognition*, 2017.

[24] T. Porter and T. Duff, "Compositing digital images," *Acm Siggraph Computer Graphics*, vol. 18, no. 3, pp. 253–259, 1984.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2014.

[26] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2017, pp. 1520–1529. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.168

[27] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman, "Learning the depths of moving people by watching frozen people," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6602–6611.

[29] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 249–256, 01 2010.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.