IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received May 11, 2020, accepted June 18, 2020, date of publication June 23, 2020, date of current version July 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004431

Image-Based Rendering for Large-Scale Outdoor Scenes With Fusion of Monocular and Multi-View Stereo Depth

SHAOHUA LIU[1,2], MINGHAO LI[1], XIAONA ZHANG[3], SHUANG LIU[3], ZHAOXIN LI[4], JING LIU[3,4], AND TIANLU MAO[4]

[1]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Institute of Electronic and Information Engineering in Guangdong, University of Electronic Science and Technology of China, Dongguan 523808, China
[3]College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang 050024, China
[4]Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Corresponding authors: Jing Liu (liujing01@ict.ac.cn) and Tianlu Mao (ltm@ict.ac.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1700905, in part by the Major Program of the National Natural Science Foundation of China under Grant 91938301, in part by the National Natural Science Foundation of China under Grant 61702482, Grant 61802109, and Grant 61532002, in part by the National Defense Equipment Advance Research Shared Technology Program of China under Grant 41402050301-170441402065, and in part by the Sichuan Science and Technology Major Project on New Generation Artificial Intelligence under Grant 2018GZDZX0034.

ABSTRACT Image-based rendering (IBR) attempts to synthesize novel views using a set of observed images. Some IBR approaches (such as light fields) have yielded impressive high-quality results on small-scale scenes with dense photo capture. However, available wide-baseline IBR methods are still restricted by the low geometric accuracy and completeness of multi-view stereo (MVS) reconstruction on low-textured and non-Lambertian surfaces. The issues become more significant in large-scale outdoor scenes due to challenging scene content, e.g., buildings, trees, and sky. To address these problems, we present a novel IBR algorithm that consists of two key components. First, we propose a novel depth refinement method that combines MVS depth maps with monocular depth maps predicted via deep learning. A lookup table remap is proposed for converting the scale of the monocular depths to be consistent with the scale of the MVS depths. Then, the rescaled monocular depth is used as the constraint in the minimum spanning tree (MST)-based nonlocal filter to refine the per-view MVS depth. Second, we present an efficient shape-preserving warping algorithm that uses superpixels to generate the warped images and blend expected novel views of scenes. The proposed method has been evaluated on public MVS and view synthesis datasets, as well as newly captured large-scale outdoor datasets. In comparison with state-of-the-art methods, the experimental results demonstrated that the proposed method can obtain more complete and reliable depth maps for the challenging large-scale outdoor scenes, thereby resulting in more promising novel view synthesis.

INDEX TERMS Image-based rendering, multi-view stereo, monocular depth estimation, view synthesis, outdoor scenes.

I. INTRODUCTION

With the increasing demand for immersive 3D content, many view synthesis methods [1]–[5] for providing realistic interactive virtual navigation have been proposed. Among these methods, image-based rendering (IBR) algorithms enable high-quality view navigation via the utilization of a set of photos of a scene, which avoid the massive cost of elaborate 3D reconstruction. Early IBR works [6], [7] synthesized images between nearby views by projecting and blending texture with a proxy geometry. These methods perform poorly if synthetic views are moving far from the input photos. With the development of underlying multi-view stereo (MVS) reconstruction methods, state-of-the-art IBR methods [2], [8]–[10] have yielded promising view synthesis results by using estimated per-view depth information to inferring the color texture of contents in novel views during rendering. However, outdoor scenes often contain large amounts of low-textured surfaces, vegetations, and sky, where 3D geometric information is difficult to be reconstructed. As a result,

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tan.

warping of such regions via estimated scene depth information is undesirable; thus, this is a challenging issue for current IBR methods.

For alleviating the view synthesis problem on large-scale outdoor scenes, the refinement of the original depth estimation has been paid more attention to IBR methods. Chaurasia *et al.* [2] proposed superpixel-based depth interpolation and shape-preserving warping for the production of plausible novel views. However, it will assign an incorrect depth if spatially neighboring superpixels are located in different objects but are of similar color. Hedman *et al.* [5] proposed a robust multi-view depth estimation method, which firstly discards erroneous depths in the initial depth maps that are estimated via plane sweep stereo and propagates the confident depth points using a first-order Poisson system; then, the interpolation results are used as near-envelope terms for Markov random field (MRF) model-based discrete label optimization. DeepBlending [10] combines the depths that are generated via two MVS methods for joint optimization in order to obtain complete and accurate depth maps. However, the missing and erroneous geometric structures in the MVS depth estimation stage still have a substantial impact on the final view synthesis.

Besides the depth refinement methods tailored for IBR problem, there are a large number of works, which formulate the depth refinement task as depth inpainting [11], [12] and depth completion [13], [14]. The achievements of deep neural networks in recent years have encouraged researchers to implicitly model problems using many training examples. DepthComp [13] realized the efficient and plausible filling of depth holes in stereo image pairs utilizing learning-bassed semantic segmentation. Zhang and Funkhouser [14] recovered the missing depth data via optimization with the surface normals estimated by a neural network. However, these methods cannot be applied to the large holes in the depth maps.

Faced with geometric inaccuracies in classical MVS methods, learning-based MVS methods [15]–[17] learned a mapping between multi-view images and 3D volumetric labeling or depth maps. However, these methods require huge training sets with ground-truth, which is infeasible for large-scale scenes in which the performances of commodity-grade depth sensors are often limited due to strong light or distance. Unsupervised methods [18]–[22] can produce coarse monocular depths without being limited to datasets with ground-truth data. Although these depths cannot be input the IBR pipeline, they can be used as a priori information for the refinement of the original MVS depth.

To improve the quality of depth estimation and view synthesis for large-scale outdoor scenes, in this work, we propose an IBR method that is based on fusion of monocular and MVS depth. The proposed method consists of a depth refinement stage and a view synthesis stage. In the depth refinement stage, we combine a learning-based monocular depth [19] with the MVS depth [23] to realize more complete and reliable depth estimation. Since the MVS depth and the monocular depth are from distributions that differ substantially in terms of scale, we present a novel layerwise mapping between the monocular depth and the MVS depth via a lookup table. Then, we propose a nonlocal algorithm that is based on a minimum spanning tree (MST) for effectively fusing the rescaled monocular depth and the MVS depth. We use semantic segmentation [24] to specify the depth of the sky. In the view synthesis stage, we build our blending solution based on a superpixel-based local shape-preserving warp. We improve the warp efficiency by using a single circumscribed triangle instead of multiple overlapping grids in the energy function minimization. An overview of our approach is presented in Fig. 1. Our main contributions are summarized as follows:

1) A lookup-table-based strategy that remaps the monocular depth to the scale of the MVS depth;
2) An MST-based algorithm for fusing the monocular depth and the MVS depth, which can fill in irregularities and large holes of MVS depth maps while preserving geometric details;
3) A complete pipeline for image-based outdoor scenes navigation, which includes a refinement method for depth estimation, and a superpixel-based shape-preserving warp for view synthesis.

The remainder of this paper is organized as follows. Section II briefly reviews the related work. Section III describes the proposed method in detail. We present our experimental results in Section IV. Finally, the conclusions of this study are presented in Section V.

II. RELATED WORK

In this section, we first review the related IBR methods. Then, we briefly review the necessary works related to depth estimation, including multi-view 3D reconstruction, depth inpainting, and monocular depth estimation.

A. IMAGE-BASED RENDERING

Early image-based rendering methods mainly include view interpolation [25], light fields [26] and Lumagraphs [7], [27]. These methods have led to some interesting applications, such as first-person hyper-lapse video [28], VR panorama [4] and commercial Google Street View. But these methods are usually not applicable to large-scale scenes under sparse capture. View-dependent texture mapping [6] uses a uniform geometry proxy to blend re-projected source images and provides a strong sense of realism in the model. This idea has been used to process wide-baseline input datasets in subsequent image-based rendering systems. The Unstructured Lumigraph [7] defines a per-pixel weighting function by combining a number of ‘fidelity criteria’. However, geometry is not always complete and accurate, especially in textureless areas. Floating textures [29] uses soft visibility in rendering to reduce the ghosting and blurring that were caused by an imprecise geometry. Sinha *et al.* [30] generated piecewise planar depth maps by solving a multilabel MRF optimization problem to improve the view interpolation performance on untextured surfaces. The non-photo realistic rendering (NPR)

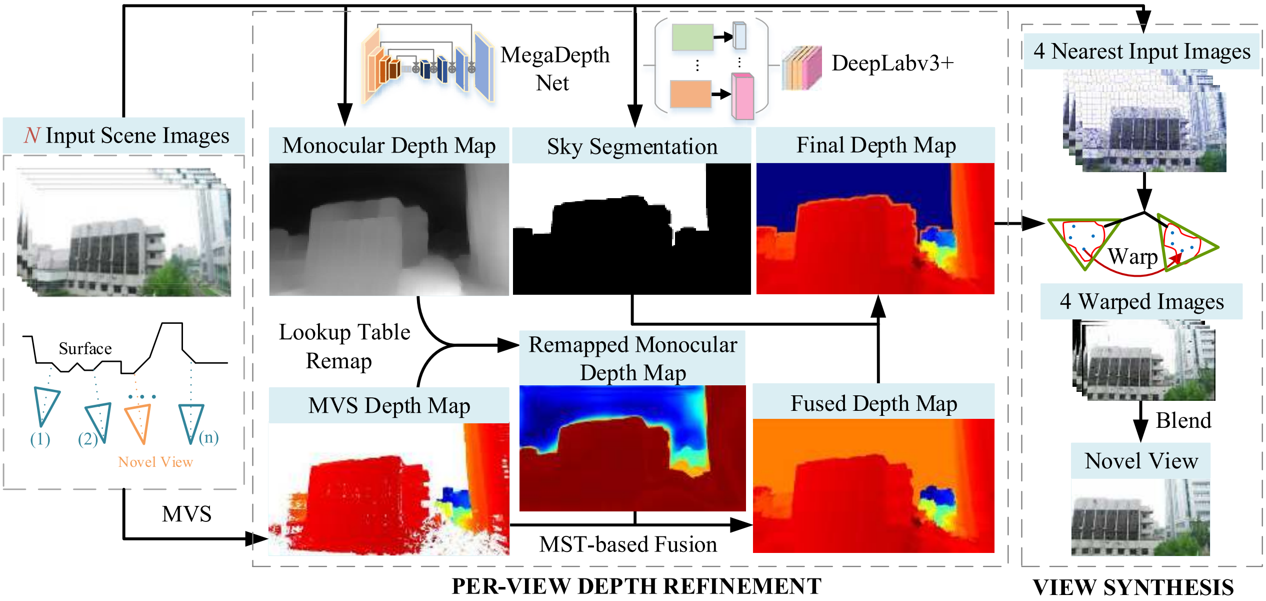

FIGURE 1. Overview of the proposed method. The input is a set of images captured from different viewpoints. The proposed depth refinement fuses monocular depth and MVS depth to improve the completeness and accuracy of per-view depth maps. Moreover, we incorporate the semantic segmentation results to detect the sky pixels. Finally, a superpixel-based shape-preserving warping is applied to synthesize the novel view.

style [31] utilizes ambient point clouds with uncertain depth to reduce artifacts. Silhouette warping [32] improves the rendering of foreground objects by protecting depth discontinuities with manual assistance. In addition, soft visibility [29], [33], superpixels [2], [8] and boundary alpha matting [34]–[36] have been used to improve the blending results at occlusion edges.

Recent IBR methods provide a more complete solution for interactive free-viewpoint navigation. Davis *et al.* [37] proposed a 2D-domain-view roaming method in pre-adjusted capture mode for fixed indoor scenes. However, complex and long-term capture is not always feasible. Chaurasia *et al.* [2] conducted depth repair and local shape-preserving warping in superpixels for wide-baseline urban datasets, thereby enabling the viewer to move far from the input cameras. To tradeoff between quality and speed, Selective IBR [8] uses a Bayesian approach to select suitable superpixels for warping. Inside-Out IBR [38] uses per-view mesh simplification and tiling to implement a real-time free-viewpoint rendering system of indoor scenes, but high-quality 3D reconstruction is derived from an RGB-D video and hundreds of images. Soft3D [9] designs a soft visibility function by retaining uncertainty in a volumetric depth-sweep and yields satisfactory rendering results across a wide variety of inputs (e.g., plenoptic and unstructured cameras and light-field video). However, it cannot be applied to large-scale scenes due to excessive memory consumption.

In recent IBR studies, in addition to being used as components, data-driven learning methods also have been used as end-to-end frameworks for view synthesis. A deep convolutional network was used for the multi-view rendering of a single object in [39], [40]. DeepStereo [41] trained two tower networks separately by using the plane sweep volume to predict the depth and color. Habtegebrial *et al.* [42] estimated depth on stereographic pairs based on a convolutional neural network (CNN) and generated texture with a forward mapping network. Although these methods yield promising results, they still suffer from blurring and low resolution, and their computational costs are too high for real-time rendering. The application of customized networks to facilitate classical algorithms is more effective. DeepBlending [10] used a convolutional neural network to compute blending weights via per-view meshes [38], thereby reducing the severity of artifacts and realizing rendering in real-time. Stereo Magnification [43] estimated multiplanar images (MPIs) at multiple depth levels using a deep network and synthesizes new views between two narrow-baseline stereo images. More recently, DeepView [44] used learned gradient descent to produce MPI and yielded high-quality view synthesis results on a light field and camera array dataset. Local light field fusion (LLFF) [45] implemented a practical image synthesis system using predicted MPI and analyzed the required light field sampling rate. However, LLFF required a parallax-limited photo collection under the guidance of an application. Based on the available proxy geometry, neural networks have also been used to support view-dependent rendering [46] and scene re-rendering under multiple appearances [47].

B. DEPTH ESTIMATION

Multi-View 3D Reconstruction: The multi-view stereo algorithms [48], [49] can reconstruct the 3D geometry from a set of photos of a scene that were captured from diverse locations or angles. For a comprehensive review of MVS, we refer the reader to Furukawa and Hernández [50]. Here, we focus on the multi-view stereo algorithms that are associated with view synthesis. A perfect 3D texture model is the best, but it can be challenging to implement using a collection of casually captured images. The related methods [48], [49] automatically reconstructed semi-dense depth maps that can be merged as a proxy in the IBR methods [7], [29]. COLMAP [23], [51] has been used in IBR methods for cameras poses estimation [5], [10], [47] and dense depth reconstruction [10]. COLMAP was also demonstrated to produce the most accurate geometry in MVS benchmark tests [52], [53]. However, depth maps that are reconstructed via COLMAP are less complete in large textureless regions due to insufficient matching features, which will cause large holes when they are applied to view synthesis, especially in outdoor scenes.

Depth Inpainting and Completion: Many heuristic methods that are based on image filtering or optimization have been proposed for filling the holes in depth maps. Most of these methods [11], [12] focus on the depth maps that are generated by depth sensors such as the Microsoft Kinect. These depth maps are typically more accurate and complete than depth maps that are obtained via MVS. Modified closing by reconstruction (McBR) [12] improved the depth maps from time-of-flight sensors by using a modified morphological closing filter. Several reconstruction iterations are required for the removal of small holes from the depth maps; however, this approach is ineffective for large holes in outdoor scenes with structures that are completely missing. Zhang and Funkhouser [14] recovered the missing sensor depth via optimization with the normals of RGB-D images that are estimated by a neural network. DepthComp [13] applied various filling strategies to classified holes in the depth maps that were classified based on a semantic segmentation prior by SegNet [54]. The results strongly depend on the accuracy of segmentation. However, DepthComp focused only on the depth map and did not consider the subsequent use of depth. Its incorrect depth filling in large holes can produce unpredictable artifacts in the view synthesis for IBR.

Monocular Depth Estimation: The increasing availability of deep learning techniques and large training datasets has led to a new generation of depth reconstruction methods that can recover the lost dimension, even from a single image. Supervised monocular depth estimation methods [55]–[57] have realized high accuracy on fixed datasets, such as NYU (indoor-only images) and KITTI (road scenes). However, these methods are limited by the available training data and have difficulty generalizing well on large-scale outdoor datasets. Among the unsupervised methods [18]–[22], MegaDepth [19] generated training data via the structure-from-motion (SFM) and multi-view stereo (MVS) methods, and constructed a large dataset from Internet photo collections. The model that was trained on this dataset exhibited strong performance in generalization to novel scenes. However, the depth maps that are produced MegaDepth [19] represent relative depths and are not view-consistent; hence, we cannot derive accurate depth values in a physical dimension from them. Thus, they cannot be directly applied in current IBR methods.

The proposed depth refinement algorithm can restore missing depth information by combining the learning-based monocular depth estimation and MVS methods. Similarly, Fácil *et al.* [58] fused CNN-based single-view and multi-view depth to improve the depth of low-parallax image sequences. Martins *et al.* [59] have demonstrated that the stereo depth leads to higher performance with the monocular estimated depth fusion.

III. OUR APPROACH

As illustrated in Fig. 1, our approach consists of two main stages: per-view depth refinement and a superpixel-based warping for view synthesis. In the following, we give the detailed introduction of depth refinement and warping method in Section III-A and Section III-B, respectively.

A. PER-VIEW DEPTH REFINEMENT

Our input is a set of photos that were captured from various viewpoints of a scene. The view synthesis quality of IBR methods is directly affected by the geometric accuracy and completeness of the MVS reconstruction. However, in an outdoor scene, there are many regions that are difficult to be reconstructed, such as non-Lambertian surfaces of buildings, trees, and sky, which poses a challenge for the available IBR algorithms. To improve the 3D reconstruction performance in the outdoor scenes, we propose the combined use of the monocular depth to repair the missing areas of the MVS depth maps. In addition, we incorporate the semantic segmentation results to specify the sky regions.

MVS Depth Preprocessing: First, We register the cameras and obtain a dense MVS depth map for each view using COLMAP, which is a general-purpose SFM and MVS pipeline. Although the geometric depth maps from COLMAP are highly accurate in most areas, there remain some scattered false depth samples on special objects (see Fig. 2b). For example, some wrong depth points appear in the sky and lush vegetation that should not exist. To filter out these outliers, we apply a combination of pruning median filters, which were proven to be effective in [5]. A pixel should be pruned if its depth in the median filtered depth map differs sufficiently from that in the original depth map (not within a factor of [0.9, 1.11]). First, we use a small (5×5) median filter to prune the sparse noise. Then, we use a larger (31×31) median filter [60] weighted by a color term ($\sigma_c = 0.033$). These pruned MVS depth maps will be used for subsequent processing steps.

Monocular Depth Remapping: We utilize a learning-based method [19] to generate monocular depth maps that ensures the geometric completeness. The monocular depth maps that are estimated by MegaDepth [19] perform well in difficult reconstruction regions by exploring context semantic information. However, the monocular depths are on a different scale with the MVS depths. The MVS depth $D_{mvs}(\mathbf{x})$ is a real value that measures the distance of the 3D point to the focal

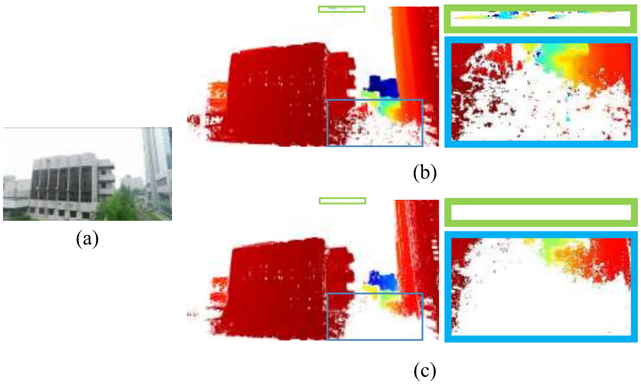

FIGURE 2. Comparison of depth map without and with pruning median filters. (a) Input color image. (b) Original geometric depth map by COLMAP. The close-ups show the wrong depth samples in sky and lush vegetation. (c) Pruned geometric depth map.

plane, and the monocular depth $G(\mathbf{x})$ is an unsigned integer value $(0, 1, \ldots, 65535)$ that decreases as the distance from the camera increases.

We design a per-view lookup table strategy for remapping the monocular depth to the scale of the MVS depth for each pixel. Since the monocular depth predicted by the neural network is not cross-view-consistent, each pair of a monocular depth map and an MVS depth map corresponds to a unique lookup table. First, we traverse all the pixels of a monocular depth map to determine the corresponding valid MVS depth values at the same coordinates. Then, all MVS depth samples are divided into 256 different levels $\mathbf{L}_{i=0,1\ldots255}$ according to the corresponding monocular depth values. The set of each level is

$$\mathbf{L}_i = \{\mathbf{x}|D_{mvs}(\mathbf{x}) > 0\}, \quad i = Z(\frac{G(\mathbf{x})}{256}), \tag{1}$$

where function $Z(\cdot)$ means taking the integer part of a float number. For each level, we choose the median as a representative MVS depth

$$d'_i = \underset{\mathbf{x}\in\mathbf{L}_i}{median}(D_{mvs}(\mathbf{x})). \tag{2}$$

Equation (2) defines a mapping between a level i and an representative MVS depth d'_i. To maintain sufficient precision, we expand this map into a lookup table that is applied to determine the discrete monocular depth by conducting a 256-level linear interpolation between neighbor-level representative depths:

$$d_{G(\mathbf{x})} = (d'_{i+1} - d'_i)(\frac{G(\mathbf{x})}{256} - i) + d'_i. \tag{3}$$

With the per-view lookup table, the monocular depth of pixel $\mathbf{x}$ can be remapped to:

$$D_{mono}(\mathbf{x}) = d_{G(\mathbf{x})}, \quad G(\mathbf{x}) \in \{0, 1, \ldots, 65535\}. \tag{4}$$

Fusion: For fusing the remapped monocular depth and the pruned MVS depth, we propose an adaptive nonlocal weighted fusion algorithm that is based on a minimum spanning tree (MST) which balances accuracy and completeness. For small holes in the MVS depth map, our algorithm uses the surrounding MVS depths to propagate a more accurate depth. In contrast, for large holes in which an entire structure is missing, we will more strongly consider the monocular depth.

For clarity, we use Ω and Ψ to denote the regions where the MVS depth is missing and is available, respectively. The overall depth fusion framework is:

$$D_{fused}(\mathbf{x}) = \begin{cases} D_{inp}(\mathbf{x}), & \mathbf{x} \in \Omega, \\ D_{mvs}(\mathbf{x}), & \mathbf{x} \in \Psi, \end{cases} \tag{5}$$

where $D_{inp}(\mathbf{x})$ is the inpainting depth, which is weighted by the monocular depth and the propagating MVS depth. The inpainting depth $D_{inp}(\mathbf{p})$ of pixel $\mathbf{p} \in \Omega$ can be predicted from the remapped monocular depth $D_{mono}(\mathbf{p})$ of pixel $\mathbf{p}$ and the MVS depth $D_{mvs}(\mathbf{q})$ of all pixels $\mathbf{q} \in \Psi$:

$$D_{inp}(\mathbf{p}) = w_{\mathbf{p}} D_{mono}(\mathbf{p}) + \sum_{\mathbf{q}\in\Psi} w_{\mathbf{q}} D_{mvs}(\mathbf{q}) \tag{6}$$

where $w_{\mathbf{p}}$ and $w_{\mathbf{q}}$ are the normalized weights of the monocular depth and the MVS depth, respectively:

$$w_{\mathbf{p}} = 1 - \sum_{\mathbf{q}\in\Psi} w_{\mathbf{q}} = \frac{\alpha}{\sum_{\mathbf{q}\in\Psi} \mathcal{S}(\mathbf{p}, \mathbf{q}) + \alpha}, \tag{7}$$

$$w_{\mathbf{q}} = \frac{\mathcal{S}(\mathbf{p}, \mathbf{q})}{\sum_{\mathbf{q}\in\Psi} \mathcal{S}(\mathbf{p}, \mathbf{q}) + \alpha}, \tag{8}$$

in which α is a constant that is used to adjust the confidence of the monocular depth, and $\mathcal{S}(\mathbf{p}, \mathbf{q})$ denotes the similarity between $\mathbf{p}$ and $\mathbf{q}$. The similarity $\mathcal{S}(\mathbf{p}, \mathbf{q})$ is expressed as

$$\mathcal{S}(\mathbf{p}, \mathbf{q}) = \mathcal{S}(\mathbf{q}, \mathbf{p}) = exp(-\frac{\mathcal{D}is(\mathbf{p}, \mathbf{q})}{\sigma}), \tag{9}$$

in an MST that is defined by [61], where σ is a constant that is used to adjust the similarity between two nodes, and $\mathcal{D}is(\mathbf{p}, \mathbf{q}) = \mathcal{D}is(\mathbf{q}, \mathbf{p})$ denotes the distance between $\mathbf{p}$ and $\mathbf{q}$ in the MST.

In [61], all pixels of a color image constitute a connected undirected graph. The edges of the graph are generated by connecting a pixel with its four neighboring pixels. The weight values of the edges are the color differences of the connected pixels. Then, the MST can be established by removing the edges with larger weights to minimize the sum of the weights. Such an MST not only provides a natural pixel similarity measure but also creates an unspecified nonlocal window for cost aggregation. Let $C_d(\mathbf{p})$ denote the matching cost for pixel $\mathbf{p}$ at disparity level d, and let $C_d^A(\mathbf{p})$ denote the aggregated cost. The aggregated cost can be computed as follows:

$$C_d^A(\mathbf{p}) = \sum_{\mathbf{q}} \mathcal{S}(\mathbf{p}, \mathbf{q}) C_d(\mathbf{q}), \tag{10}$$

where $\mathbf{q}$ denotes all other pixels in the MST/image except $\mathbf{p}$. The calculation of the aggregation cost of each pixel from other pixels one by one is slow. However, the similarity distance between two nodes in MST can be calculated cumulatively as MST is traced from the leaf nodes to the root node.

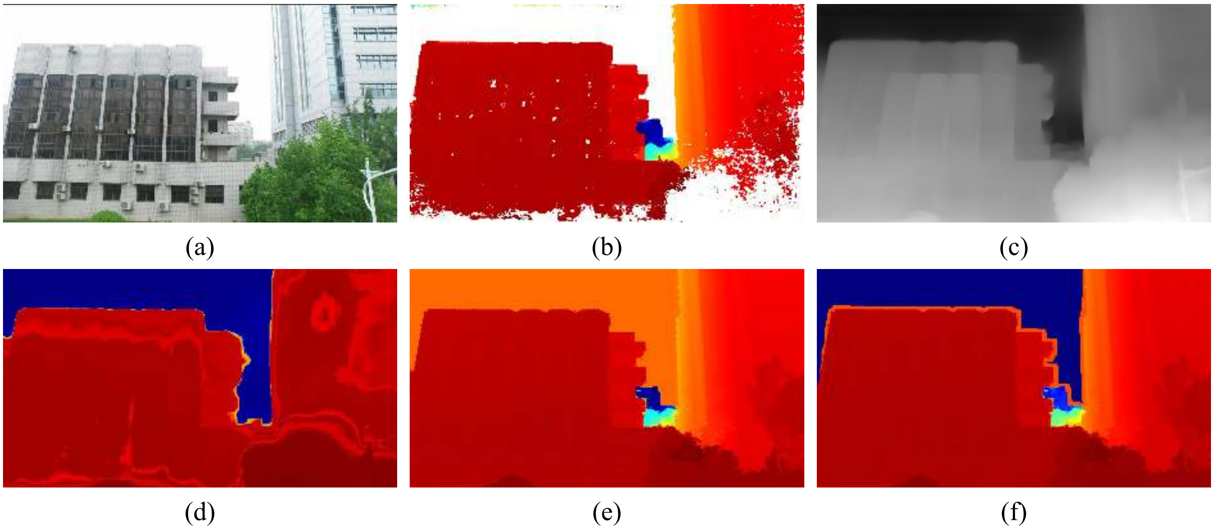

FIGURE 3. (a) Input image. (b) Monocular depth map predicted by neural networks. (c) Semi-dense depth map estimated by MVS, (d) Monocular depth map remapped to the scale of MVS depth. (e) Fused depth map by using MVS and monocular depth maps. (f) Final refined depth map after filling up the sky depth.

Hence, the calculation of the aggregation cost $C_d^A(\mathbf{p})$ for each node requires two addition/subtraction operations and three multiplication operations in [61].

We will not repeat the details of the fast calculation of the cost aggregation with the MST, but we will describe how to quickly fuse the depth based on Equation (10). Let $C_d(\mathbf{q}) = 1$ and $C_d(\mathbf{q}) = 0$ denote whether or not, respectively, there is available MVS depth at pixel $\mathbf{q}$. We can transform Equation (10) into the following:

$$\mathcal{J}^A(\mathbf{p}) = \sum_{\mathbf{q}\in\Psi} \mathcal{S}(\mathbf{p},\mathbf{q}). \tag{11}$$

Similarly, we can calculate the aggregated MVS depth at pixel $\mathbf{p}$:

$$D^A_{mvs}(\mathbf{p}) = \sum_{\mathbf{q}\in\Psi} \mathcal{S}(\mathbf{p},\mathbf{q}) D_{mvs}(\mathbf{q}). \tag{12}$$

By combining Equation (11) (12) and (6) (7) (8), we can calculate the inpainting depth by:

$$\begin{aligned} D_{inp}(\mathbf{p}) &= w_{\mathbf{p}} D_{mono}(\mathbf{p}) + \sum_{\mathbf{q}\in\Psi} w_{\mathbf{q}} D_{mvs}(\mathbf{q}) \\ &= \frac{\alpha}{\mathcal{J}^A(\mathbf{p}) + \alpha} D_{mono}(\mathbf{p}) + \frac{D^A_{mvs}(\mathbf{p})}{\mathcal{J}^A(\mathbf{p}) + \alpha}. \end{aligned} \tag{13}$$

As demonstrated in [61], we can obtain the $D^A_{mvs}(\mathbf{p})$ and $\mathcal{J}^A(\mathbf{p})$ of all pixels $\mathbf{p} \in \Omega$ efficiently by two cumulative calculations from the root node to the leaf node and then from the leaf node to the root node. There are two constants in our algorithm: $\sigma = 0.06$, $\alpha = 0.00001$ in all our experiments, and α can be increased if the monocular depth has higher confidence.

Depth synthesis for sky. Photos that were captured outdoors contain sky regions in most cases. The proposed lookup table strategy cannot correctly convert the monocular depth in the sky regions because no reasonable MVS depth sample is available. We use the DeepLabv3+ [24] to identify the sky regions Φ, and we specify the depth of the sky to be 200 percent of the max MVS depth D_{max} to obtain the final fused depth map:

$$D_{final}(\mathbf{x}) = \begin{cases} 2D_{max}, & \mathbf{x} \in \Phi, \\ D_{fused}(\mathbf{x}), & otherwise. \end{cases} \tag{14}$$

With the development of semantic segmentation models based on deep learning, it is not difficult to identify the rough sky area in the scene. However, the detailed contours that are obtained via semantic segmentation are not sufficiently good enough (see Fig. 4a); hence, foreground objects may be incorrectly classified as sky regions. These fake sky depth samples will produce artifacts at the edges of the sky in the subsequent image warping (see Fig. 4d). In superpixel segmentation, the superpixels around the edges may contain both sky and foreground (see Fig. 5a). These superpixels will be warped via the foreground's depth when sky depth is not synthesized, as shown in Fig. 4b. Once these superpixels are filled with sky depth, they will be warped as the sky. The foreground pixels they contained will be assigned to unreasonable depth and cause visible foreground artifacts.

Therefore, we can "protect" the foreground content by reducing the area of the sky depth, and thus tolerate semantic segmentation with less precise contour. Here we reduce the area of the sky mask by applying a morphological erosion operation with a diameter of approximately 3% of the image diagonal (see Fig. 4e). The result in Fig. 4f demonstrates that the erosion operation is beneficial for improving the image quality at the occlusion edge. Fig. 3f shows a final refined depth map.

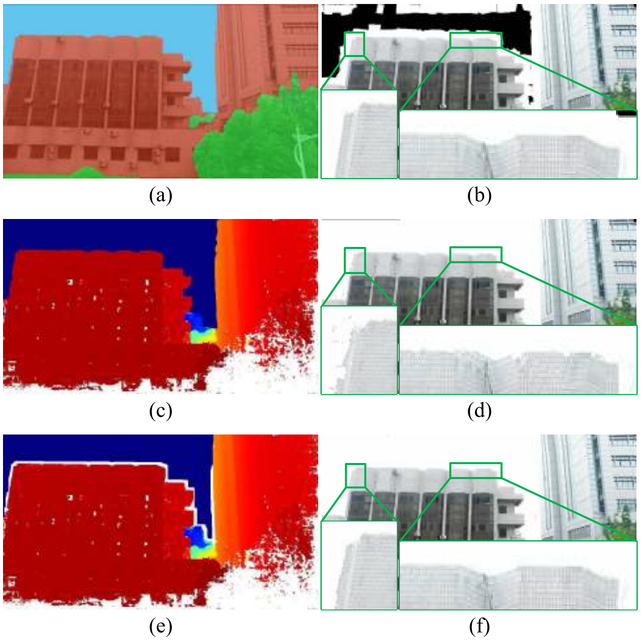

FIGURE 4. Semantic segmentation of the sky and comparison of results using different sky masks. (a) Original semantic segmentation (sky, vegetation, and others) by DeepLabv3+ [24]. (c) and (e) Depth synthesis of sky on MVS depth map by original and eroded sky mask, respectively. (b), (d), and (f) View synthesis results by MVS depth map, (c), and (e), respectively.

B. WARPING AND RENDERING

Direct reprojection of texture pixels for a novel view will result in disturbing artifacts. The state-of-the-art IBR algorithms [8], [10] attempt to alleviate this problem by using per-view geometric structures, which can realize satisfactory visual quality. However, [10] requires full-resolution meshing and computationally expensive rendering. Superpixel-based warping [8] can yield plausible view synthesis results if the superpixels contain sufficiently many reconstructed points. Shape-preserving warp [2] is regarded as one of the highest quality superpixel-based methods and is applied to a poorly reconstructed nonplanar structure in [8]. Therefore, for complex outdoor scenes, we adopt a variational warping technique that is similar to [2].

In contrast to the overlapping multiple mesh grids that are used for each superpixel in [2], we calculate a circumscribed triangle T as the warping grid for each superpixel to reduce the computational costs. For a superpixel S_k, the constraints are composed of two energy terms: the reprojection energy at each pixel and the shape-preserving energy, namely, the energy for preserving the shape of the superpixel during the warp. In the warp of a source image I to a novel view N, each superpixel satisfies these two energy terms in a least-squares sense. For S_k, we denote the vertices of its circumscribed triangle T by $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and each pixel that belongs to superpixel S_k by $\mathbf{x} \in S_k$. The barycentric coefficients of pixel $\mathbf{x}$ in triangle T are (α, β, γ):

$$\mathbf{x} = \alpha\mathbf{v}_1 + \beta\mathbf{v}_2 + \gamma\mathbf{v}_3. \tag{15}$$

The circumscribed triangle will change after the warp of each superpixel. We denote the new vertices of circumscribed triangle T by $(\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \tilde{\mathbf{v}}_3)$. The reprojection energy at pixel $\mathbf{x}$ is defined by:

$$E_p(\mathbf{x}) = \|\alpha\tilde{\mathbf{v}}_1+\beta\tilde{\mathbf{v}}_2+\gamma\tilde{\mathbf{v}}_3 - C_N\circ(C_I^{-1}\circ(\mathbf{x}, D(\mathbf{x}))^T\|^2, \tag{16}$$

where C_I^{-1} is the back-projection operator of image I, C_N is the projection of novel view N, and $D(\mathbf{x})$ is the depth value of $\mathbf{x}$. This energy term measures the distance between each pixel position of the warped triangle T and the reprojected location $\tilde{\mathbf{x}}$. The shape-preserving energy of a superpixel is defined by:

$$\begin{aligned} a &= (\mathbf{v}_3 - \mathbf{v}_1)^T(\mathbf{v}_2 - \mathbf{v}_1)/\|(\mathbf{v}_2 - \mathbf{v}_1)\|, \\ b &= (\mathbf{v}_3 - \mathbf{v}_1)^T\mathcal{R}_{90}(\mathbf{v}_2 - \mathbf{v}_1)/\|(\mathbf{v}_2 - \mathbf{v}_1)\|, \\ E_s(T) &= \|\tilde{\mathbf{v}}_3 - (\tilde{\mathbf{v}}_2 + a(\tilde{\mathbf{v}}_1 - \tilde{\mathbf{v}}_2)) + b\mathcal{R}_{90}(\tilde{\mathbf{v}}_1 - \tilde{\mathbf{v}}_2)\|^2, \end{aligned} \tag{17}$$

where $\mathcal{R}_{90}$ is a counterclockwise 90° rotation. For each circumscribed triangle with vertices $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, this energy term measures its shape distortion after the warp. The overall warp energy function for each superpixel S_k is as follows:

$$E(S_k) = E_s(T) + \sum_{\mathbf{x}\in S_k} E_p(\mathbf{x}). \tag{18}$$

We minimize $E(S_k)$ for each superpixel to solve the warped optimization problem. The unknown values at the energy minimum are three new vertices $(\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \tilde{\mathbf{v}}_3)$ of the triangle. We use a 6×1 vector instead of the x- and y-coordinates of three unknown vertices to construct $E(S_k)$ into a sparse matrix system, and we solve the vector value at the minimum of the system. Then, we use the triangle interpolation to obtain the texture of each warped superpixel in the novel view.

Our rendering consists of three steps: first, we select and warp the four nearest input images close to the novel camera position. Next, we reproject the median depth (see Fig. 5b) of a superpixel into the novel view for the depth test to remove superpixels behind the camera. The warped superpixels of each image are separately rendered via the depth test. Finally, we blend the warped superpixel images to synthesize the novel view by selecting the color values of the pixels with the highest weights (see Fig. 6). The blending weights are computed from the angle penalty in [32] at each pixel.

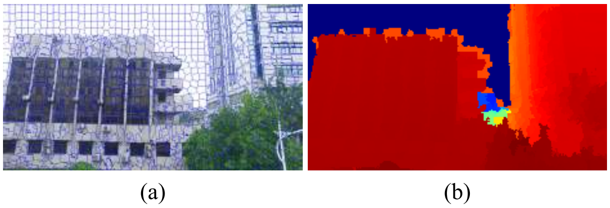

FIGURE 5. (a) Superpixel oversegmentation. (b) The median depth of each superpixel.

IV. RESULTS AND COMPARISONS

In this section, we evaluate our method on a wide variety of datasets, which include eight public datasets and

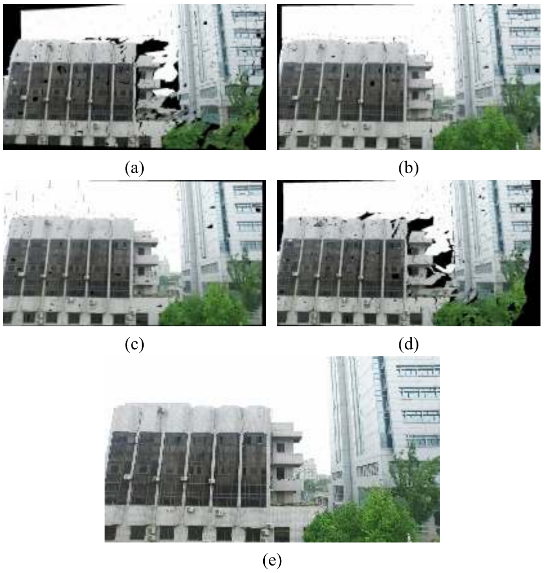

FIGURE 6. Warped images and the final view synthesis result. (a) (b) (c) (d) are four warped images from the neighboring cameras. (e) Synthesized result.

three datasets captured by ourselves. *Aquarium-20*, *Hugo-1*, *Tree-18*, *Museum-1*, *Museum-2*, *Street-10*, *Poche* and *Bridge* are obtained from previous studies [2], [10], [32]. Each contains 13-106 photos of an urban scene that were captured by DSLR cameras. The *GTAV* is a synthetic dataset from [17], which consists of 120 image sequences of urban streetscape with rendered ground-truth depth information. *Campus*, *Xumi* and *Guya* were captured by us in large-scale outdoor scenes. We captured several high-resolution videos using a consumer drone (DJI MAVIC Pro) for each scene. Then, 20-30 images were subsampled from each video stream. *Campus* contains lush foreground vegetation, reflective surfaces, and distant buildings, and *Guya* and *Xumi* contain large nature scenes with grottoes and hills.

A. EVALUATION OF THE DEPTH REFINEMENT RESULTS

First, we present qualitative results of our depth refinement on various outdoor datasets. As shown in Fig. 7, the missing regions of the original depth maps are well repaired by our refinement algorithm. The depth maps from the MVS reconstruction present various missing depths in these datasets. In *Guya* and *Tree-18*, the original depth information is comparatively complete. In *Xumi* and *Museum-1*, the depth-missing regions are located mainly at the occlusion edges. The depth information of the lush vegetation near the camera is almost completely absent in *Museum-1* and *Campus*. The monocular depth maps that were estimated by [19] have complete and reasonable outlines. However, we can identify errors in the monocular depth maps (color images) that were converted by the lookup table. For example, the monocular

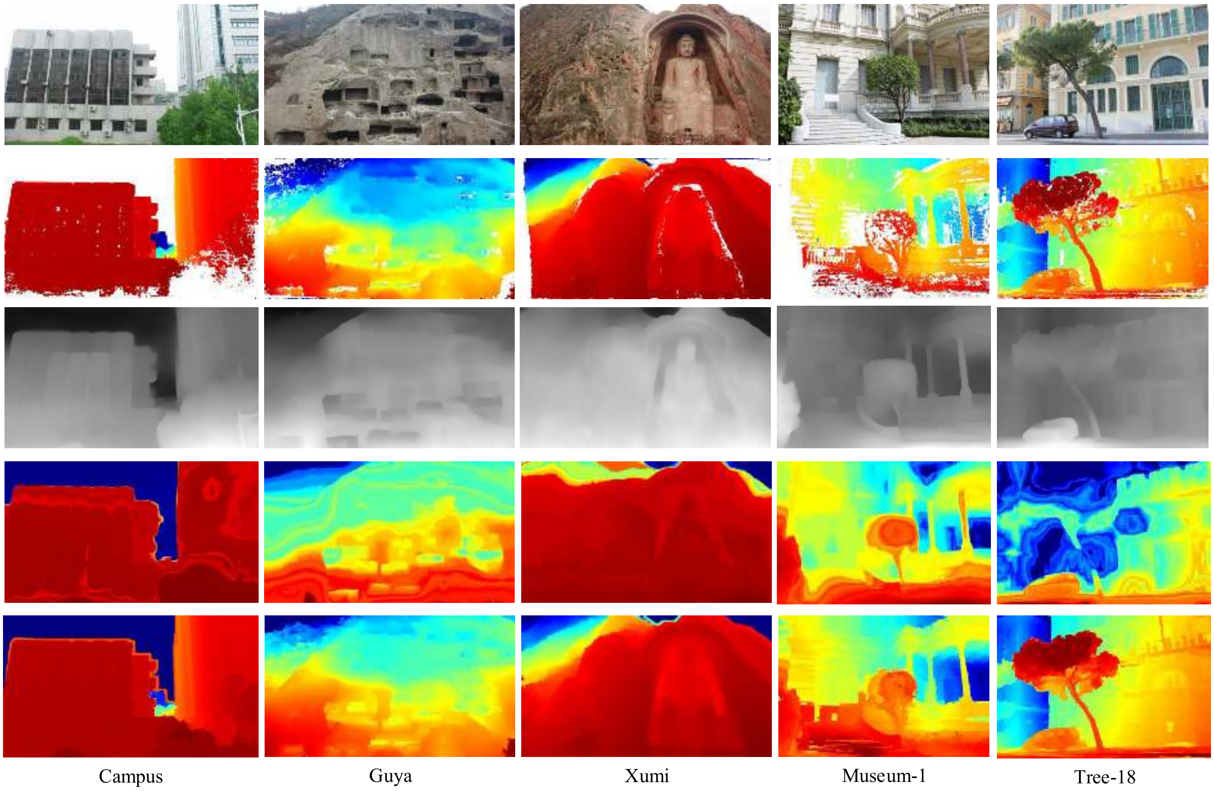

FIGURE 7. Depth maps from five datasets. From top to bottom: input images, the original depth maps by COLMAP, monocular depth maps estimated by MegaDepth [19], monocular depth maps remapped with the lookup table and our final refined depth map with the sky segmentation.

TABLE 1. Errors of remapped monocular depth by different methods on various datasets.

Dataset	Median factor	Linear function	Ours
Aquarium-20	0.1014	0.1566	**0.0782**
Museum-1	0.1124	0.1370	**0.0882**
Museum-2	0.1086	0.1219	**0.0733**
Tree-18	0.2063	0.5020	**0.1529**
Hugo-1	0.4242	1.1813	**0.2248**
Campus	0.1848	3.8095	**0.1479**
Guya	0.2421	0.4147	**0.1304**
Xumi	0.1977	5.4064	**0.1665**
GTAV_MVS	0.6152	1140.8	**0.4941**
GTAV_GT	0.5810	9.5757	**0.4707**

Error metric: Abs Rel (absolute relative error)

depth is sometimes incorrect (foreground objects in *Guya* and *Tree-18*) or even absent (distant buildings in *Campus*). The final depth maps that were refined by our algorithm generate promising results and perform well on various datasets.

The core component of our monocular depth remapping algorithm is a hierarchical lookup table. A numerical comparison of three remapping methods is presented in Table 1. Monocular depth estimation methods [18]–[21] computed the median ratio between the predicted monocular depth and the ground-truth depth as a scale factor in each view for evaluation. Similarly, the monocular depth can be remapped to the scale of the MVS depth by using a median factor. We also present the remapped results that are obtained by using a L_2 fitted linear function in the second column.

In the first eight rows of Table 1, we use three methods to remap the monocular depth map to the scale of MVS depth and compute the error of remapped monocular depth relative to the available MVS depth points. In the last two rows of Table 1, we use the MVS depth (GTAV_MVS) and the ground-truth depth (GTAV_GT) as the references for remapping, respectively, and then compute the error between the remapped monocular depth and the ground-truth depth. Our method outperformed the other approaches. Linear functions are sensitive to large outlier noise, especially in outdoor datasets with large depth ranges. The median factor will tend to be larger (smaller) if more foreground (background) information is missing from the MVS depth map. Our lookup table associates two types of depth samples by pixel location and remaps the monocular depth via multiple layers to reduce the impact of the missing MVS depth.

Finally, we qualitatively (Fig. 8 and Fig. 9) and quantitatively (Table 2) demonstrate that our depth fusion algorithm outperforms the inpainting methods [12], [13], [62].

For fairness, we report the results of DepthComp that were obtained using the more advanced DeepLabv3+ [24] as a segmentation component instead of SegNet [54]. The DeepLabv3+ network is pretrained on the *Cityscapes* [63] dataset. FMM [62] is a gradient-propagation-based algorithm that has been applied to color image inpainting successfully, which repairs erroneous pixels by weighting the available values in their neighborhoods. We use this method as a baseline method for depth inpainting. Ablation experiments are conducted by removing the monocular depth map ($\alpha = 0.0$) from our fusion algorithm.

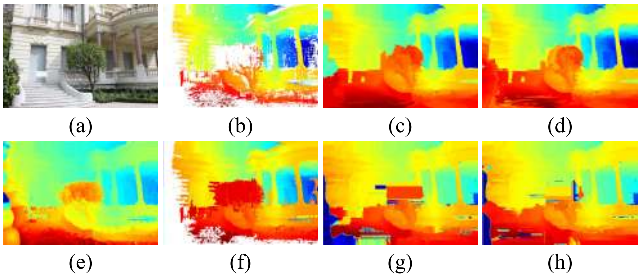

FIGURE 8. Depth map completion results on the Museum-1 dataset. (a) Input image. (b) Original MVS depth map by COLMAP [23]. (c) and (d) Our depth refinement result without monocular depth ($\alpha = 0.0$) and with monocular depth ($\alpha = 0.00001$). (e), (f), (g), and (h) Depth inpainting result by FMM [62], McBR [12], DepthComp [13] + SegNet [54], and DepthComp [13] + DeepLabv3+ [24].

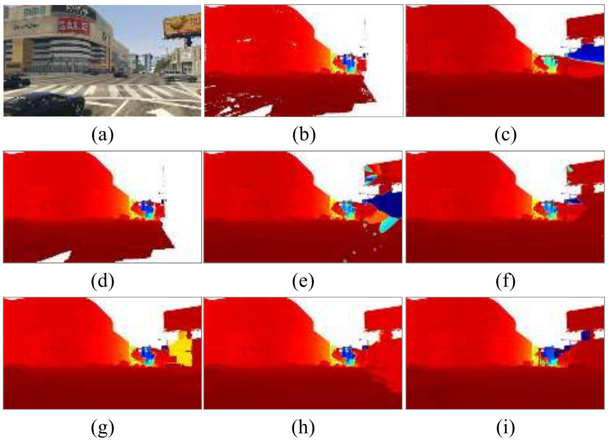

FIGURE 9. Depth map completion results on the GTAV dataset. (a) Input image. (b) Original MVS depth map by COLMAP [23]. (c), (d), (e), and (f) Depth inpainting result by FMM [62], McBR [12], DepthComp [13] + SegNet [54], and DepthComp [13] + DeepLabv3+ [24]. (g) and (h) Our depth refinement result without monocular depth ($\alpha = 0.0$) and with monocular depth ($\alpha = 0.00001$). (i) Ground truth.

As shown in Fig. 8, our depth refinement method with monocular depth produced complete and reliable depth information in the left shrub, where the MVS depth is almost completely missing. The other methods and our method without monocular depth propagated the incorrect depth information in that region. Fig. 9 shows the depth completion results of various methods on the *GTAV* dataset. We removed the depth of the sky based on the ground truth to better visualize the depth map. Most methods perform well on the small holes in the original COLMAP depth map. However, only we produce reasonable depth information in large missing areas. Table 2 presents the numerical results. We use RMSE and MAE to measure the accuracy of the depth map, and PBRE to measure the completeness of the depth inpainting.

B. COMPARISONS OF THE VIEW SYNTHESIS RESULTS

First, we show how our depth refinement improves the view synthesis results. In Fig. 10, we visually compared

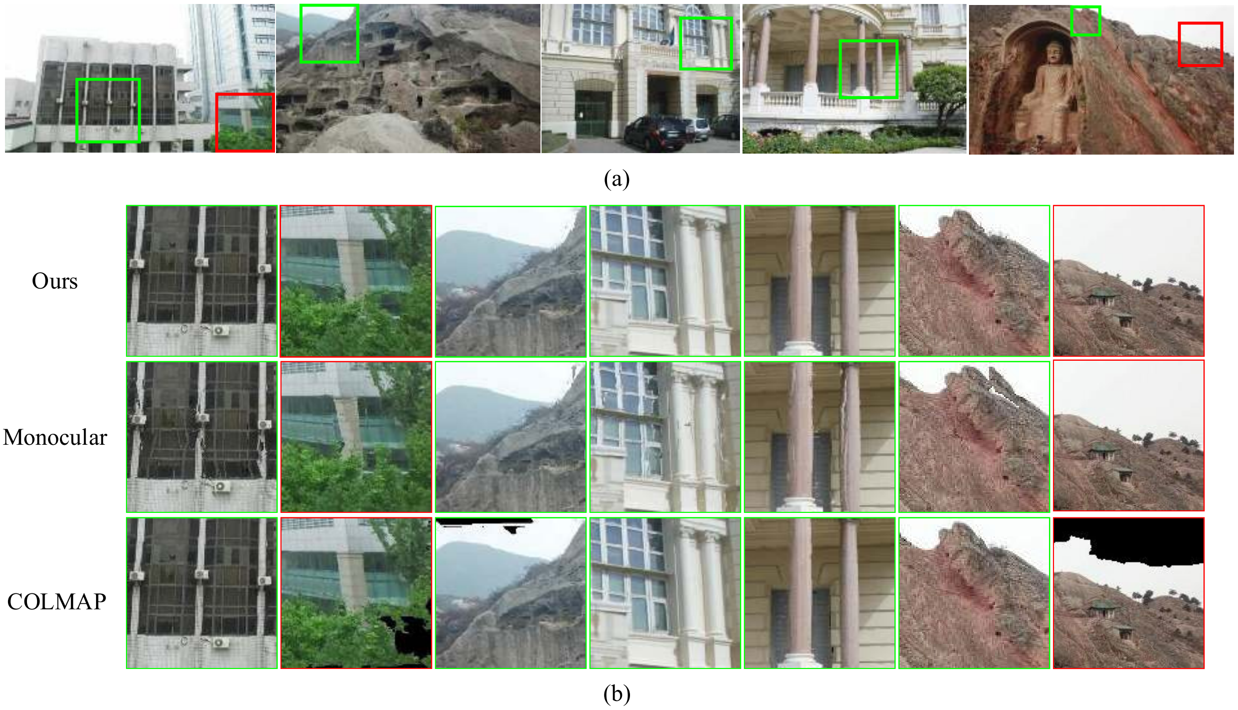

FIGURE 10. View synthesis results by using different depth maps in Campus, Guya, Aquarium-20, Museum-1, and Xumi. (a) Results by using final refined depth maps. (b) From top to bottom: cropped images of synthesized views based on depth maps generated by ours, remapped monocular depth, and original COLMAP depth, respectively.

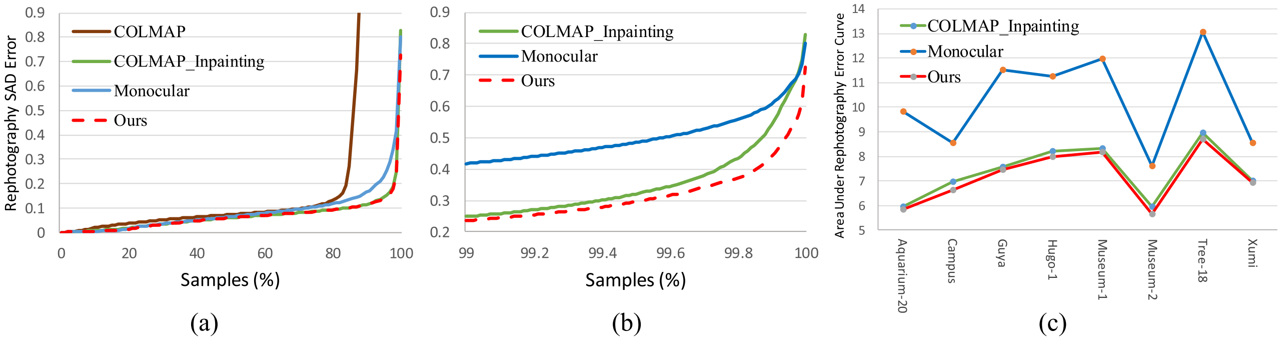

FIGURE 11. Rephotography comparison, how well different depth maps can reconstruct held-out input images. (a) The cumulative distribution of SAD errors by using original COLMAP depth, COLMAP depth with inpainting, remapped monocular depth, and ours on Campus dataset. The x-axis is the percentage of pixels with an error smaller than a threshold (y-axis). (b) We compare the error curves of (a) at the high error parts (99%-100%). (c) The area under the rephotography SAD error curves (0%-100%) on eight datasets (Aquarium-20, Hugo-1, Tree-18, Museum-1, Museum-2, Campus, Guya, and Xumi).

the view synthesis results that were obtained using three depth maps and our superpixel-based warping method. Due to the difficulty of MVS reconstruction, there are visible holes in the vegetation and sky (red cropped regions) of the synthesized view using original COLMAP depth. The results that were obtained using the remapped monocular depth perform well in difficult reconstruction areas but cause distortion artifacts (green cropped regions). Our approach effectively combines the advantages of both depth maps. The results that were obtained using our final refined depth not only fill the holes but also reduce the artifacts.

We further conduct a quantitative comparison using Virtual Rephotography [64]. The rephotography SAD error has been adopted in recent end-to-end IBR methods [5], [10] for quantitative evaluation. We calculate the rephotography SAD error from the "shiftable L_1 error" [10], which is the minimum L_1 distance that is obtained comparing a 7×7 color patch around each ground-truth image pixel with the same-sized output patches of the synthetic view. The output patch is allowed to shift up to ± 2 pixels in the x- and/or y-direction around the source pixel.

A rephotography evaluation is obtained by using our, COLMAP and monocular depth maps in Fig. 11a.

TABLE 2. Results of depth inpainting by different methods on GTAV dataset.

Method	RMSE	MAE	PBRE-0.05	PBRE-0.07	PBRE-0.10
COLMAP [23]	774.293	10.8627	11.94%	10.87%	9.96%
FMM [62]	759.530	9.7215	10.08%	8.68%	7.36%
McBR [12]	825.922	10.5788	11.34%	10.28%	9.34%
DepthComp [13] +SegNet [54]	956.043	11.2208	9.91%	8.59%	7.39%
DepthComp [13] +DeepLabv3+ [24]	917.128	10.8915	**9.72%**	8.39%	7.16%
Ours($\alpha = 0.0$)	723.855	9.4396	9.80%	8.32%	6.93%
Ours	**703.832**	**9.4173**	9.73%	**8.26%**	**6.88%**

Error metric: RMSE (root mean square error), MAE (mean absolute error), PBRE-0.05, PBRE-0.07, PBRE-0.10 (percentage of bad pixels with absolute relative error greater than 0.05, 0.07, 0.10). Lower is better for all error metrics.

TABLE 3. The sum of the area under rephotography error curves (AUC) over eight datasets by different window sizes and shiftable lengths.

Patch Size	Shiftable Length	COLMAP_ Inpainting	Monocular	Ours
5 × 5	±2	55.81379	77.57284	**54.44593**
	±1	60.96495	90.17267	**59.60735**
	0	79.59808	114.47753	**78.36583**
7 × 7	±2	58.90011	82.34624	**57.44466**
	±1	62.91692	93.26034	**61.52862**
	0	78.75605	113.89062	**77.46437**
15 × 15	±2	63.31546	89.52072	**61.86216**
	±1	65.45173	97.39725	**63.95193**
	0	76.76642	112.50197	**75.32561**

Datasets: *Aquarium-20*, *Hugo-1*, *Tree-18*, *Museum-1*, *Museum-2*, *Campus*, *Guya* and *Xumi*.

The performance of using COLMAP depth maps is the worst, which resulted in large holes due to incomplete depth maps. For fairness, we fill these holes on synthesized views by an image inpainting [62], named COLMAP_Inpainting, which performs well for areas with a homogeneous color like the sky. In this way, the curves of COLMAP_Inpainting and ours look almost coincident. However, the artifacts by human perception mainly occur in the high error parts. We plot the high error parts (the percentage of 99-100) of three curves in Fig. 11b. The result using our depth maps is significantly better than the other two methods. Fig. 11c presents the area under rephotography error curves (AUC) (the percentage of 0-100) of three methods on eight datasets: *Aquarium-20*, *Hugo-1*, *Tree-18*, *Museum-1*, *Museum-2*, *Campus*, *Guya* and *Xumi*.

In order to verify the performance by different patch sizes or shiftable lengths, we calculated the sum of AUC values over eight datasets in Table 3. We selected 5, 7, 15 patch sizes and 2, 1, 0 shiftable lengths on images with a resolution of 960 × 640 or 960 × 540. It can be seen that results using our depth maps get smaller error values than the other two.

Then, we present a quantitative comparison with other depth refinement methods focusing on IBR. Depth Synthesis (DS) [2] interpolated depth samples at the missing of PMVS [48] reconstruction based on superpixels' similarity. DeepBlending [10] combined the depths of two MVS reconstruction: COLMAP [23] and RealityCapture [65], but its refined depth maps are still missing where both MVS fail. Since DeepBlending [10] incorporates different MVS methods to improve the completeness of depth maps while ensuring accuracy, we can regard the DeepBlending depth as a better MVS source. We applied the DS algorithm and our depth refinement on three MVS sources and then evaluated them via the Virtual Rephotography, as shown in Table 4. For faithful comparison, we use the same superpixel-based warping method for all results.

Since the quantitative comparison of view synthesis requires unified camera parameters, we extract them from the COLMAP sparse reconstruction of all datasets provided by DeepBlending [10]. In addition, COLMAP depth maps and DeepBlending depth maps also come from the datasets provided by DeepBlending [10]. We import the sparse reconstruction and input images into PMVS program to generate a dense point cloud and project it into each input view to generate the PMVS depth maps.

As shown in Table 4, compared to DS [2], using our depth refinement algorithm can obtain better view synthesis results on various MVS sources. The results of using our scheme (COLMAP+Ours) are not much different from that of using DeepBlending depth refinement method on most datasets, even better on *Museum-2*. More importantly, our algorithm can still improve the DeepBlending refined depth further while DS can't. It should be noted that the completeness and accuracy of depth map both contribute to the quality of view synthesis results. Both ours and DS improve the completeness of depth maps by synthesizing plausible depth in missing areas. However, incorrect synthesized depth will introduce noise, which sometimes makes the results worse, such as DeepBlending+DS.

In Fig 12, we show the view synthesis results of using four depth maps on *Museum-1*. The result of DeepBlending produces a white blur at the window (top of the green cropping) due to lack of depth. While the results of COLMAP+Ours and PMVS+DS have a reasonable texture, there are some breakages at the balusters (green croppings). In the area around the pillar (red croppings), the ranking of view quality from good to bad is DeepBlending+Ours, DeepBlending, COLMAP+Ours, PMVS+DS. In the blue croppings, the result of PMVS + DS is the worst, while the other three are almost the same. The proposed refinement scheme (COLMAP+Ours) achieves significantly better results than Chaurasia *et al.* [2] (PMVS+DS), and our results are similar to DeepBlending [10]. It worths noting that we can further apply our refinement algorithm on depth maps outputted by DeepBlending, giving rise to the best results.

In addition, one advantage of our algorithm is speed. DeepBlending optimizes per-pixel depth by minimizing the global photoconsistency cost. It takes 45 minutes for DeepBlending to perform depth refinement on *Creepy Attic* (249 images at 1228 × 816), while our algorithm only takes 2.8 minutes. DS takes 1.8 minutes on the same inputs.

TABLE 4. The AUC scores of rephotography by using different MVS depth sources and refinements.

	COLMAP [23]			DeepBlending [10]			PMVS [48]		
	Original	+DS [2]	+Ours	Original	+DS [2]	+Ours	Original	+DS [2]	+Ours
Aquarium-20	5.6145	5.6171	5.5174	5.5007	5.5007	**5.4929**	7.3990	7.2796	7.1694
Museum-1	7.5755	7.5720	7.3857	7.3831	7.3831	**7.3337**	14.6927	14.5493	14.3044
Museum-2	5.7924	5.8056	**5.6138**	5.7280	5.7285	5.7188	10.4321	10.3613	10.1516
Hugo-1	7.8987	7.8931	7.6482	7.6196	7.6197	**7.5828**	13.1645	12.9360	12.7270
Tree-18	8.9437	8.9501	8.7497	8.6778	8.6777	**8.6677**	12.4193	12.0485	11.8998
Street-10	11.8213	11.7849	11.3741	11.1443	11.1448	**11.1067**	20.2672	19.8715	19.7704
Poche	18.0341	18.0285	17.7628	**17.6251**	17.6257	17.6523	31.3669	31.4661	31.2771
Bridge	21.9942	21.9144	21.9517	20.9257	20.9320	**20.6325**	—	—	—

The images of all datasets are resampled to 1920 pixels wide. The window size and shiftable length for SAD error calculation are 7 × 7 and ±2, respectively. The depth maps of some views (1,18,21,79,91,93-95,98) reconstructed by PMVS on *Bridge* (106 images) are completely empty so that the comparison is canceled.

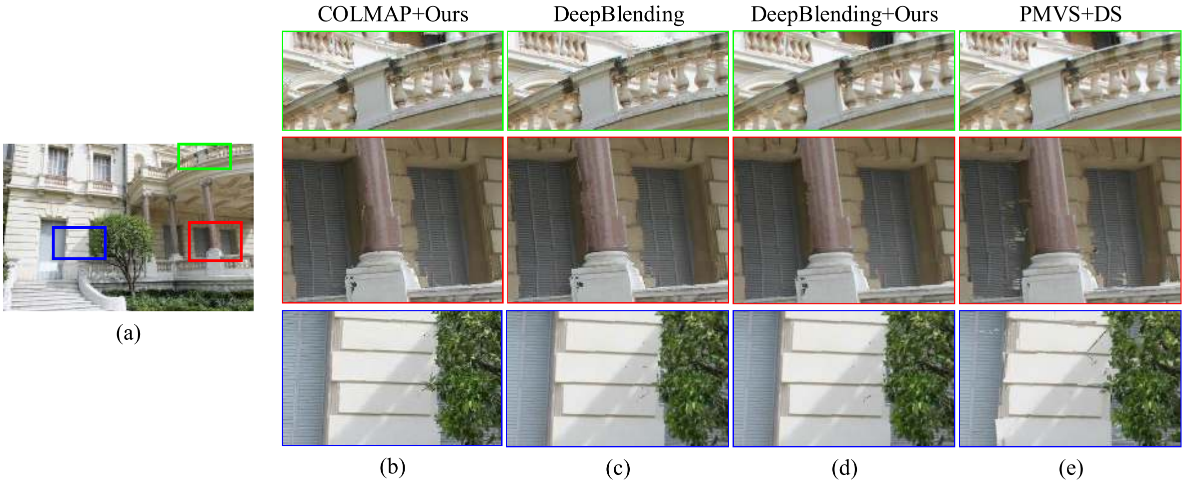

FIGURE 12. View synthesis results by using different MVS sources and depth refinements on Museum-1. All results are obtained by using our warping and rendering components. (a) and (b) Full novel view and cropped images from our solution (COLMAP+Ours). (c),(d), and (e) The cropped images using depth maps from DeepBlending [10], DeepBlending [10] with Our refinement (DeepBlending+Ours) and Chaurasia *et al.* [2] (PMVS+DS), respectively.

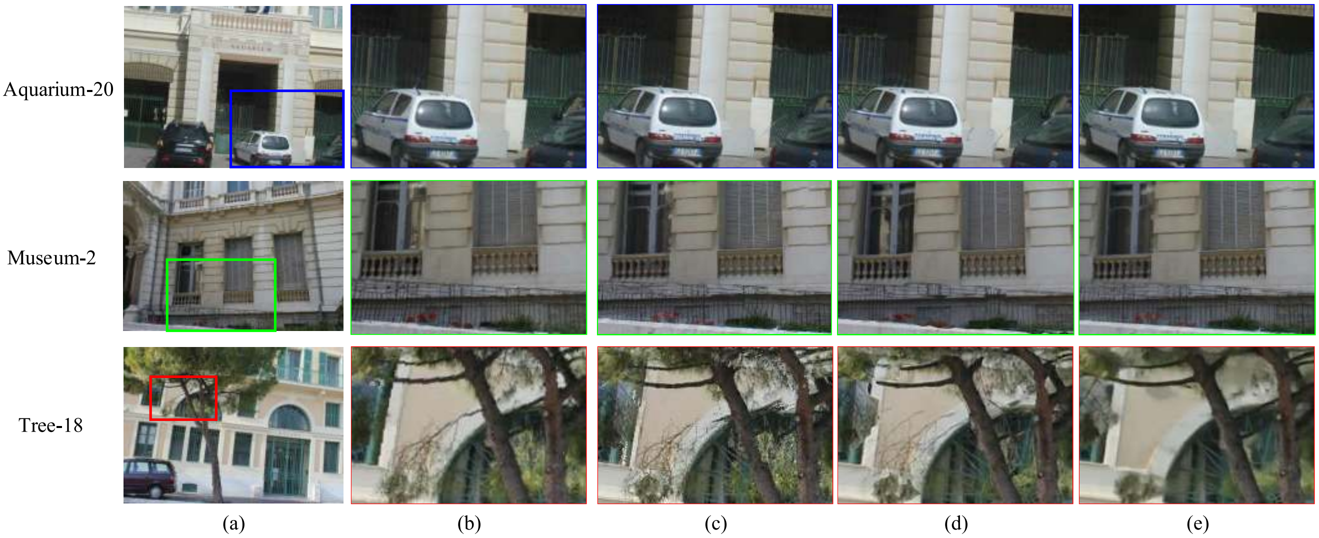

FIGURE 13. Comparison of view synthesis on three urban datasets. (a) and (b) Full novel views and cropped images from our solution. (c), (d), and (e) The cropped images of Selective-IBR [8] based on superpixels warp, ULR [7] improved by soft visibility [29], and DeepBlending [10], respectively. The cropped images of other methods come from the supplemental material of paper [10].

Finally, we show the view synthesis results from three urban datasets by our complete IBR pipeline and three competing approaches [7], [8], [10] in Fig. 13. These approaches have already shown their superiority compared with the unstructured lumigraph rendering [7], manually defined silhouettes warp [32] and global textured mesh [65]. Compared to Selective-IBR [8], which is also based on superpixels warp, the results of our method avoid blurring in trees (*Tree-18*) and reduce foreground warp artifacts (*Aquarium-20*). There are some aliasing artifacts on our cropped image from the *Tree-18*, which may be caused by using a large superpixel size in our warping.

V. CONCLUSIONS

We present a complete IBR method that effectively uses output of neural networks to refine the MVS depth. Better scene geometry is obtained by combining the monocular and multi-view stereo depth. Semantic segmentation is applied to identify the sky region, avoiding manual corrections [32]. Compared to propagating depth from the boundary purely, our synthesized depth is more reasonable for large holes and avoids excessive smoothness. It is demonstrated that our refined depth map is capable of improving the reconstructions of a variety of challenging outdoor datasets and achieves view synthesis as good as the competing methods.

A primary limitation of our algorithm is that the pre-trained monocular estimation and semantic segmentation network models may not generalize well on unfamiliar scene content and reduce the accuracy of fused depth. Currently we use the depth estimation network of MegaDepth [19] and semantic segmentation network of DeepLabv3+ [24]. The performance of the proposed depth refinement and view synthesis method may be improved as more advanced networks are integrated. Another limitation is the temporal flicker when rendering novel viewpoints continuously due to the per-view shape-preserving warping. We show this defect in the supplementary video. This could be addressed by exploring a temporal weighted blending strategy such as DeepBlending [10].

REFERENCES

[1] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, Jul. 2006, doi: 10.1145/1141911.1141964.

[2] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Trans. Graph.*, vol. 32, no. 3, p. 30, Jun. 2013, doi: 10.1145/2487228.2487238.

[3] A. Collet, "High-quality streamable free-viewpoint video," *ACM Trans. Graph.*, vol. 34, no. 4, p. 69, Jul. 2015, doi: 10.1145/2766945.

[4] R. Anderson, "Jump: Virtual reality video," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–13, Nov. 2016 doi: 10.1145/2980179.2980257.

[5] P. Hedman, S. Alsisan, R. Szeliski, and J. Kopf, "Casual 3D photography," *ACM Trans. Graph.*, vol. 36, no. 6, p. 1–15, Nov. 2017, doi: 10.1145/3130800.3130828.

[6] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1996, pp. 11–20, doi: 10.1145/237170.237191.

[7] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured Lumigraph rendering," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 2001, pp. 425–432, doi: 10.1145/383259.383309.

[8] R. O. Cayon, A. Djelouah, and G. Drettakis, "A Bayesian approach for selective image-based rendering using superpixels," in *Proc. Int. Conf. 3D Vis.*, Oct. 2015, pp. 469–477.

[9] E. Penner and L. Zhang, "Soft 3D reconstruction for view synthesis," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 235-1–235-11, 2017. doi: 10.1145/3130800.3130855.

[10] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Trans. Graph.*, vol. 37, no. 6, p. 257:1–257:15, Dec. 2018, doi: 10.1145/3272127.3275084.

[11] F. Qi, J. Han, P. Wang, G. Shi, and F. Li, "Structure guided fusion for depth map inpainting," *Pattern Recognit. Lett.*, vol. 34, no. 1, pp. 70–76, Jan. 2013.

[12] M. A. Gardu no-Ram on, I. R. Terol-Villalobos, R. A. Osornio-Rios, and L. A. Morales-Hernandez, "A new method for inpainting of depth maps from time-of-flight sensors based on a modified closing by reconstruction algorithm," *J. Vis. Commun. Image Represent.*, vol. 47, pp. 36–47, 2017.

[13] A. Atapour-Abarghouei and T. Breckon, "Depthcomp: Real-time depth image completion based on prior semantic scene segmentation," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2017, pp. 1–13.

[14] Y. Zhang and T. Funkhouser, "Deep depth completion of a single RGB-D image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 175–185.

[15] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "DeepMVS: Learning multi-view stereopsis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2821–2830.

[16] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon, "Dpsnet: End-to-end deep plane sweep stereo," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Dec. 2018, pp. 1–7.

[17] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 767–783.

[18] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[19] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from Internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050.

[20] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[21] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019.

[22] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proc. 33rd AAAI Conf. Artif. Intell. (AAAI)*, 2019, pp. 8001–8009.

[23] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 501–518.

[24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 801–818.

[25] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. 20th Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1993, pp. 279–288, doi: 10.1145/166117.166153.

[26] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1996, pp. 31–42, doi: 10.1145/237170.237199.

[27] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1996, pp. 43–54, doi: 10.1145/237170.237200.

[28] J. Kopf, M. F. Cohen, and R. Szeliski, "First-person hyper-lapse videos," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 78-1–78-10, 2014, doi: 10.1145/2601097.2601195.

[29] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. De Aguiar, N. Ahmed, C. Theobalt, and A. Sellent, "Floating textures," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 409–418, 2008.

[30] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1881–1888.

[31] M. Goesele, J. Ackermann, S. Fuhrmann, C. Haubold, R. Klowsky, D. Steedly, and R. Szeliski, "Ambient point clouds for view interpolation," *ACM Trans. Graph.*, vol. 29, no. 4, p. 1–6, Jul. 2010, doi: 10.1145/1778765.1778832.

[32] G. Chaurasia, O. Sorkine, and G. Drettakis, "Silhouette-aware warping for image-based rendering," in *Proc. 22nd Eurographics Conf. Rendering*. Cham, Switzerland: Eurographics Association, 2011, pp. 1223–1232, doi: 10.1111/j.1467-8659.2011.01981.x.

[33] K. Pulli, H. Hoppe, M. Cohen, L. Shapiro, T. Duchamp, and W. Stuetzle, ''View-based rendering: Visualizing real objects from scanned range and color data,'' in *Rendering Techniques*, J. Dorsey and P. Slusallek, Eds. Vienna, Austria: Springer, 1997, pp. 23–34.

[34] R. Szeliski and P. Golland, ''Stereo matching with transparency and matting,'' in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 517–524.

[35] 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, pp. 600–608, Aug. 2004, doi: 10.1145/1015706.1015766.

[36] S. W. Hasinoff, B. Kang, and R. Szeliski, ''Boundary matting for view synthesis,'' in *2004 Conf. Comput. Vis. Pattern Recognit. Workshop*, Jun. 2004, p. 170.

[37] A. Davis, M. Levoy, and F. Durand, ''Unstructured light fields,'' in *Comput. Graph. Forum*, vol. 31. Hoboken, NJ, USA: Wiley, 2012, pp. 305–314.

[38] P. Hedman, T. Ritschel, G. Drettakis, and G. Brostow, ''Scalable inside-out image-based rendering,'' *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–11, Nov. 2016, doi: 10.1145/2980179.2982420.

[39] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, ''View synthesis by appearance flow,'' in *Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 286–301.

[40] A. Dosovitskiy, J. Tobias Springenberg, M. Tatarchenko, and T. Brox, ''Learning to generate chairs, tables and cars with convolutional networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 692–705, Apr. 2017.

[41] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, ''Deep stereo: Learning to predict new views from the World's imagery,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5515–5524.

[42] T. Habtegebrial, K. Varanasi, C. Bailer, and D. Stricker, ''Fast view synthesis with deep stereo vision,'' 2018, *arXiv:1804.09690*. [Online]. Available: http://arxiv.org/abs/1804.09690

[43] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, ''Stereo magnification: Learning view synthesis using multiplane images,'' in *Proc. SIGGRAPH*, 2018, pp. 1–7.

[44] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2367–2376.

[45] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, ''Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,'' *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, Jul. 2019.

[46] J. Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Nießner, ''IGNOR: Image-guided neural object rendering,'' 2018, *arXiv:1811.10720*. [Online]. Available: http://arxiv.org/abs/1811.10720

[47] M. Meshry, D. B Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla, ''Neural rerendering in the wild,'' 2019, *arXiv:1904.04290*. [Online]. Available: http://arxiv.org/abs/1904.04290

[48] Y. Furukawa and J. Ponce, ''Accurate, dense, and robust multiview stereopsis,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.

[49] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, ''Multi-view stereo for community photo collections,'' in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[50] Y. Furukawa and C. Hernández, ''Multi-view stereo: A tutorial,'' *Found. Trends Comput. Graph. Vis.*, vol. 9, nos. 1–2, pp. 1–148, 2015.

[51] J. L. Schonberger and J.-M. Frahm, ''Structure-from-Motion revisited,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.

[52] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, ''Tanks and temples: Benchmarking large-scale scene reconstruction,'' *ACM Trans. Graph.*, vol. 36, no. 4, p. 78, 2017, doi: 10.1145/3072959.3073599.

[53] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, ''A multi-view stereo benchmark with high-resolution images and multi-camera videos,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2538–2547.

[54] V. Badrinarayanan, A. Kendall, and R. Cipolla, ''SegNet: A deep convolutional encoder-decoder architecture for image segmentation,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[55] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, ''Deep ordinal regression network for monocular depth estimation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[56] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, ''Structured attention guided convolutional neural fields for monocular depth estimation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3917–3925.

[57] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, ''Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1043–1053.

[58] J. M. Facil, A. Concha, L. Montesano, and J. Civera, ''Single-view and multi-view depth fusion,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1994–2001, Oct. 2017.

[59] D. Martins, K. Van Hecke, and G. De Croon, ''Fusion of stereo and still monocular depth estimates in a self-supervised learning context,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 849–856.

[60] Q. Zhang, L. Xu, and J. Jia, ''100+ times faster weighted median filter (WMF),'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2830–2837.

[61] Q. Yang, ''A non-local cost aggregation method for stereo matching,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1402–1409.

[62] A. Telea, ''An image inpainting technique based on the fast marching method,'' *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004, doi: 10.1080/10867651.2004.10487596.

[63] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, ''The cityscapes dataset for semantic urban scene understanding,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[64] M. Waechter, M. Beljan, S. Fuhrmann, N. Moehrle, J. Kopf, and M. Goesele, ''Virtual rephotography: Novel view prediction error for 3D reconstruction,'' *ACM Trans. Graph.*, vol. 36, no. 1, pp. 1–11, Feb. 2017, doi: 10.1145/2999533.

[65] C. RealityCapture. (2016). *Realitycapture*. Accessed: Apr. 4, 2019. [Online]. Available: https://www.capturingreality.com

SHAOHUA LIU received the B.S. and M.Sc. degrees from Zhejiang University, in 1998 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, in 2006. He is currently an Associate Professor with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, China. His research interests include telecommunications engineering and software engineering.

MINGHAO LI is currently pursuing the M.Sc. degree with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include 3-D vision and view synthesis.

XIAONA ZHANG is currently pursuing the M.Sc. degree with the College of Computer and Cyber Security, Hebei Normal University. Her research interests include computer vision and deep learning, specifically for depth estimation.

SHUANG LIU is currently pursuing the M.Sc. degree with the College of Computer and Cyber Security, Hebei Normal University. Her research interests include computer vision and deep learning, specifically for semantic segmentation.

ZHAOXIN LI received the Ph.D. degree in computer application technology from the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. From 2011 to 2013, he was a Visiting Student with the Department of Neurological Surgery, University of Pittsburgh, USA. He was a Research Assistant and a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2014 to 2015 and from 2018 to 2019, respectively. He is currently an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include 3-D vision and view synthesis.

JING LIU was born in Hebei, China, in 1985. He received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2016. He is currently a Lecturer with Hebei Normal University. His main research interests include augmented reality, medical image analysis, and computer vision.

TIANLU MAO received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2009. She is currently working as an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics and computer vision.