Optics and Lasers in Engineering 147 (2021) 106726

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Free-viewpoint image based rendering with multi-layered depth maps

Honglin Yuan [a,b,*], Remco C. Veltkamp [b]

[a] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, Jiangsu Province 210044, China
[b] Department of Information and Computing Sciences, Utrecht University, Utrecht 3584CC, The Netherlands

ARTICLE INFO

Keywords:
Depth refinement
View synthesis
Image based rendering

ABSTRACT

Our aim is to give free-viewpoint photo-realistic rendering of real indoor scenes, using a sparse collection of RGB-D image as input. Image based rendering (IBR) is an effective way to achieve both realism and interactivity. However, there are several challenges for IBR: misalignment of object boundaries between color-and-depth image pairs often leads to ghost contours; projection errors result in the visibility failure; and useless and redundant input views often produce blurring images. To address these issues, we propose a pixel-to-pixel multi-view depth refinement method to produce pixel-accurate alignment between color-and-depth image pairs, and an adaptive view selection approach to avoid choosing redundant or useless input views. Furthermore, we propose a layered 3D warping to handle occluded elements. These components are designed to work together, reducing visual artifacts and providing plausible free-viewpoint synthesized images. The evaluation results indicate that our method achieves good performance on a wide variety of challenging scenes and performs best among popular IBR algorithms designed for dynamic scenes.

1. Introduction

Recent years have witnessed increasing demand on reproducing realistic virtual versions of real scenes for many applications, such as 3D video, free-viewpoint television (FTV) and virtual navigation of museums, libraries and games. One promising approach that provides photo-realistic imagery of real scenes is image based rendering (IBR) [1,2]. To synthesize a photo-realistic image, IBR performs 3D warping that maps color and depth information from a reference view to a novel view without full 3D reconstruction. It allows users to interactively control their viewpoints and synthesize novel views from arbitrary positions. Since the running time of IBR depends mainly on the display resolution, it does not require as much computing power as geometry-based view synthesis approaches which often require a per-view mesh with more than a million vertices to render novel views. However, various visual artifacts like ghost contours, holes and occlusion often appear in synthesized images produced by IBR.

Ghost contours are mainly caused by the misalignment of boundaries between a color image and its corresponding depth map. There are plenty of studies to correct misalignment by erasing edge-transitional regions[3,4] or smoothing depth edges[5]. Nevertheless, these methods are more likely to introduce new visible artifacts, for they may remove useful information when erasing misalignment. On the contrary, we introduce a pixel-to-pixel multi-view depth refinement algorithm to take advantage of useful information and produce pixel-accurate alignment between color-and-depth image pairs. In addition, our depth refinement method is able to fill missing depth information with the consideration of photometric and geometric consistency among multiple images.

Redundant and useless input views often lead to blurry or incomplete synthesized images. Previous studies select input images by comparing angles or distances between input and target views [6,7], and use a fixed number of input images for rendering. Therefore, they may fail to choose enough input views or choose incorrect and redundant views. In order to avoid such cases, we propose a novel view selection method providing an adaptive number of well-chosen input views to fill holes in synthesized images.

In the blending process, visibility is often solved by the Z-buffer method that only recovers the front-most pixels [8]. However, the Z-buffer approach fails to solve the visibility problem caused by depth or projection errors. As a result, when these errors exist, objects in the foreground may be occluded by objects in the background in the synthesized image. To address this issue, we divide the depth map into layers and apply 3D warping to synthesize images on each layer with a switching median filter to avoid the loss of visible information and over-smoothing. Since layered depths have the ability to represent occluded elements, our approach is better in dealing with the visibility problem.

Our main contributions are summarized as follows:

• A novel depth refinement algorithm that respects photo-consistency and edge preservation to correct misalignment between color-and-depth image pairs and fill missing depth information.

* Corresponding author.
E-mail address: h.yuan@uu.nl (H. Yuan).

https://doi.org/10.1016/j.optlaseng.2021.106726
Received 6 January 2021; Received in revised form 15 June 2021; Accepted 20 June 2021
Available online 3 July 2021

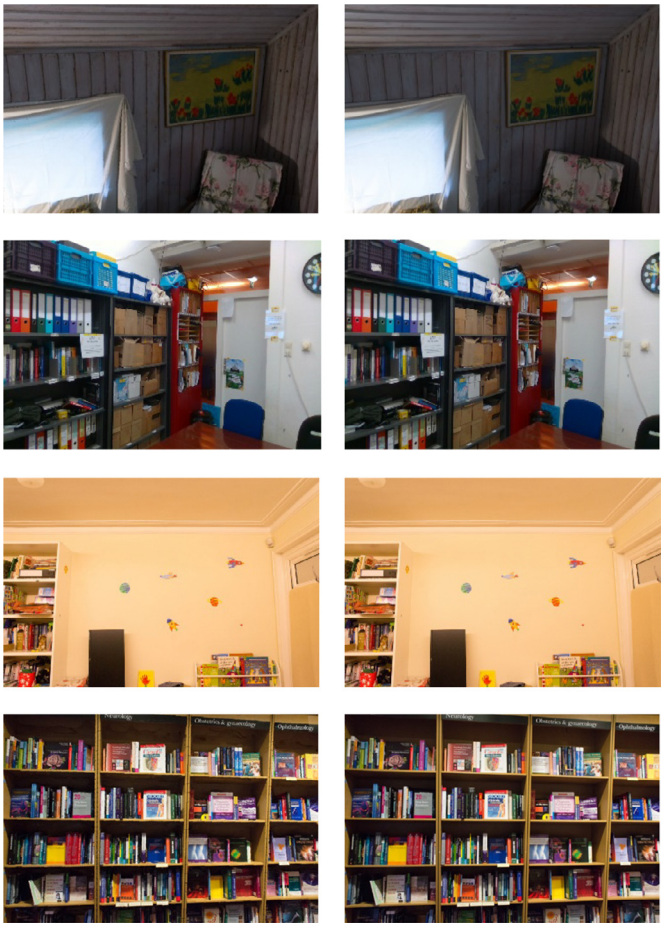


Fig. 1. Qualitative comparison between simulated images (first column) and ground truth images (second column) on datasets of Attic, Study room, Playroom and Reading corner.

• A novel adaptive view selection approach that effectively avoids selecting redundant and useless input views to improve the quality of synthesized images and the rendering speed.

• A novel rendering algorithm providing high-quality free-viewpoint synthesized images, which is based on layered 3D warping to handle occluded elements and lower the rendering complexity.

We have applied our algorithm on a variety of complex indoor scenes, as shown in Fig. 1, demonstrating that our method provides high-quality results, and can significantly improve the peak signal to noise ratio (PSNR) compared to previous works.

2. Related work

Image based rendering has been an active research area over a long period and a thorough review of it can be found in [9]. Here we only discuss approaches that are most closely related to our work.

2.1. Object boundary alignment

The importance of maintaining the alignment of object boundaries between color-and-depth image pairs has been known for many years [10,11]. Zitnick et al. [10] use color oversegment to detect object boundaries and then use neighboring Markov Random Filed to reduce artifacts at these areas. Similarly, Wang et al. [12], Buyssens et al. [13] and Ortiz-Cayon et al. [14] divide the image into superpixels to preserve depth discontinuities and then project each superpixel to a virtual view by a local shape-preserving warping to improve the blending quality. Hedman et al. [15] combine two multi-view stereo methods to produce new depth maps which respect occlusion edges. Penner et al. provide a soft model [16] to ensure robustness to depth uncertainty. However, these approaches do not consider geometric consistency among input images and still suffer from silhouette flattening and inaccurate occlusion edges.

More recently, deep learning-based approaches have been applied to synthesize virtual views [17–20]. Srinivasan et al. [21] train deep learning pipelines to predict the local geometry for blending, which is helpful to avoid boundary misalignment. Flynn et al. [22] learn gradient descent using multiplane images (MPIs), improving performance on challenging scene features such as object boundaries. Ni et al. [23] use unsupervised learning approach by a forward-backward warping process to make full use of geometry consistency, which is also helpful to achieve boundary alignment between color and depth images. However, in the current state, these methods still suffer from high computational costs. Besides, they are not suitable for small datasets collected from a large range of viewpoints.

2.2. Hole filling

There have been a lot of works [3–5] that improve the quality of synthesized images by filling holes. Schmeing et al. [24] use the inpainting approach [25] to fill holes. Solh et al. [26] use a hierarchical pyramid-like method to detect pixels of holes from lower resolution estimates of the synthesized image. They then fill holes use background information. Similarly, Dai et al. [6] also use the hierarchy idea to explore the depth distribution of neighboring pixels around each hole. Based on the distribution, they choose a number of pixels from the background and use them for hole filling. However, when there is no background information available near a hole, these approaches have poor performance. Li et al. [2] use multiple reference views to fill holes, which has some similarity with ours. Nevertheless, the number of input views used in their method is fixed, which may lead to hole filling failure when the chosen view are useless or redundant. Instead, we use an adaptive number of input views to makes sure the given virtual view can be sufficiently covered.

2.3. Visibility

To solve visibility for synthesized images, many blending methods [3,27–29] often use the Z-buffer algorithm. Hedman et al. [15] use a fuzzy depth test based on Z-buffer to blend multiple images. Dai et al. [6] defines a threshold to blend images with the similar idea like Z-buffer. However, these methods are generally unable to remove background information wrongly appearing in the foreground, which is caused by projection errors or incorrect depths. Unlike these approaches, we propose a layered 3D warping approach to resolve visibility, which can effectively reduce the loss of foreground information and remove background information that wrongly appears in the foreground.

3. Overview

Our goal is to achieve free-viewpoint rendering even in regions where a global 3D reconstruction of the scene has missing or inaccurate data for both weak and strong computing power devices.

High-quality IBR depends on the precise depth values and pixel-accurate alignment of object boundaries between color-and-depth image pairs. This is because inaccurate depth values and misalignment often lead to various visual artifacts, such as ghost contours. Unlike previous methods [7,27], which only aim to correct the misalignment of object boundaries between color-and-depth image pairs, we aim to correct misalignment and fill missing depth information at the same time. Inspired by the idea of Patchmatch stereo[30] that in natural stereo pairs relatively large regions of pixels can be modeled by approximately the same plane, we propose a pixel-to-pixel multi-view depth refinement method to refine depth maps. With the consideration of photo-consistency and

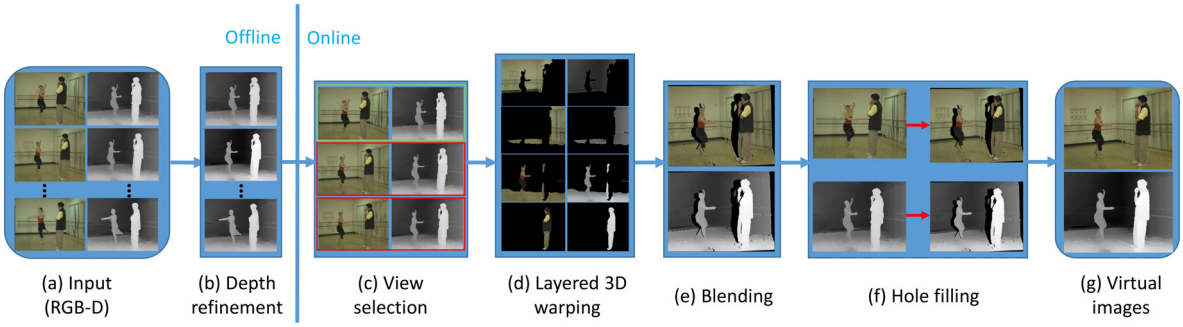


Fig. 2. Overview of our algorithm. (a) The input of our method are color-and-depth image pairs. (b) The initial depth maps are refined to achieve better alignment between object boundaries of color-and-depth image pairs. (c) Images with small view angles, short distances and large overlaps are chosen as input images. (d) We divide the depth map into layers and perform 3D warping on each layer. (e) The synthesized images are blended together. (f) Holes in the synthesized image are filled with other input images, generating the final virtual images (g).

edge preservation among multiple images, our approach is able to generate high-quality depth maps.

Even with high-quality depth maps, the synthesized image may still have holes caused by the lack of input images. However, increasing the number of input images is likely to introduce redundant or useless images. These additional images can sometimes be worse than a number of well-chosen images, as they may blur synthesized images. Besides, the more input images are chosen, the more computation time is required. To overcome these problems, we present an adaptive view selection algorithm which chooses input images based on angles, distances and overlaps between two views to avoid selecting useless and redundant images. In the rendering process we use a variable number of input images to synthesize the virtual image to lower the rendering complexity and improve the quality of synthesized images.

When blending input images, the Z-buffer method is often used to solve visibility. The intuition behind it is that closer objects occlude farther objects. However, it is not sufficient to achieve high quality for IBR. To further improve the quality of synthesized images, we divide the depth map into layers, and then apply the 3D warping on each layer to produce the virtual image. Furthermore, we present a switching median filter to fill missing information in the layered synthesized image to avoid the loss of visible information and over-smoothing problem. After that, we blend these virtual images together to produce the final synthesized image.

Combining the novelties above, our pipeline works as follows: During offline processing, we employ a pixel-to-pixel multi-view depth refinement approach to improve the quality of initial depth maps by generating pixel-accurate alignment of object boundaries between color-and-depth image pairs and filling missing depth information (see Section 4.1). During online processing, to avoid blurring images, we select input images not only based on angles and distances but also on the overlap between the input and virtual views in the query dataset (see Section 4.2). We then apply a layered 3D warping that can better handle occluded elements to synthesize virtual images (see Section 4.3). Finally, we iteratively fill holes in the synthesized image with a variable number of input images (see Section 4.2). Fig. 2 shows the pipeline of our work.

4. Free-viewpoint image based rendering

4.1. Depth refinement

High-quality depth maps are necessary for consistent rendering. However, the depth map generated by 3D sensors often has inaccurate depth values and seldom aligns object boundaries with its corresponding color image, as illustrated in Fig. 3(a). There, the background color pixel A is wrongly assigned with a foreground depth value, and in the transitional region the color pixel B which should be assigned a foreground depth value turns to have a background depth value. Fig. 3(b) shows the refined depth map we aim to produce, where color pixel A and B are assigned with correct depth values. To achieve this goal, we propose a pixel-to-pixel multi-view depth refinement approach with the consideration of photometric and geometric consistency among pixels. Our matching cost function C is defined as,

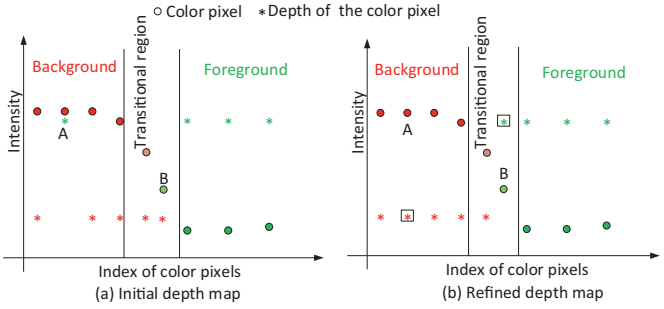


Fig. 3. Problems in the depth map and refinement results. (a) The background color pixel A has an incorrect depth and the foreground color pixel B mismatches the background depth in the transitional region. (b) The incorrect depth and misalignment are corrected after depth refinement.

$$C(i) = C_{pixel}(i) + C_{patch}(i), \tag{1}$$

where $C_{pixel}(i)$ and $C_{patch}(i)$ emphasize photo-consistency and edge preservation for the pixel i, respectively.

The photo-consistency $C_{pixel}(i)$ for the pixel i is measured by projecting it to other images, where we compare the color and gradient similarities. It is defined by:

$$C_{pixel}(i) = \lambda\vec{C}(\mathrm{x_i}) - \vec{C}(\mathrm{x_r}) + (1-\lambda)\nabla\vec{C}(\mathrm{x_i}) - \nabla\vec{C}(\mathrm{x_r}), \tag{2}$$

where x_i is the pixel we calculate cost for in the target image and x_r is the corresponding pixel of x_i in the reference image. $\vec{C}(x_i)$ and $\vec{C}(x_r)$ are the RGB components of pixel color x_i and x_r, respectively. Also, $\vec{C}(\mathrm{x_i}) - \vec{C}(\mathrm{x_r})$ and $\nabla\vec{C}(\mathrm{x_i}) - \nabla\vec{C}(\mathrm{x_r})$ indicate the color and gradient differences, respectively. λ is a measure parameter. We set $\lambda = 0.9$ in all the experiments. For a target image, we first base on distances and angles between the target and reference views to select reference color images. The maximum number of selected reference images is ten. Next, we iteratively project pixels in the reference image to the target image and only save the cost value of the front-most pixel. As it is possible that several pixels in the reference image can be projected to the same position, all the projected depth values are needed to be compared. In this way, we are able to avoid obtaining high cost values for correct depths.

The edge preserving term $C_{patch}(i)$ encourages the resulting depth map to have pixel-accurate alignment with its corresponding color im-

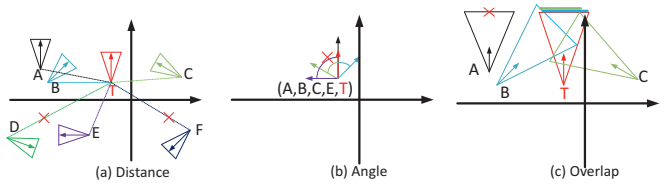


Fig. 4. View selection pipeline. (a) A, B, C, D, E, and F are input views and T is the target view. We first select a cluster of images which have short distances between the target and input views. (b) From the cluster of images, we select a subset of images with small angles they have with the target view. (c) Based on the overlap between the target and input views, we remove views having no overlaps with T.

age.

$$C_{patch}(d_i) = \frac{1}{N}\sum_{q\in W_i} e^{-\vec{C}(x_i)-\vec{C}(x_q)}, \tag{3}$$

where W_i denotes a small (3×3) patch centered on pixel i which is chosen empirically and q is the neighbor pixel of i. N is the size of the patch (3×3). $\vec{C}(x_i) - \vec{C}(x_q)$ computes the L1 norm between the RGB colors of i and q.

In the depth refinement process, we first select pixels that need to be modified based on the cost value. If the cost value of pixel p is bigger than the average cost of a $(n\times n)$ patch centered on it, we search the patch to find the lowest cost (pixel q) in the patch. This is because correct depth values have low matching costs that are computed with the consideration of photometric and geometric relationships among pixels. Next, we replace the depth and cost of p with q's, for spatial neighboring pixels are likely to have similar depth values. We run this process until all the pixels are compared. The comparison process is interleaved with the depth refinement. That is propagating good depth values to neighbors, if the costs are smaller than those of their neighbors.

After propagation, we filter unusual depths with a weighted median filter [31] which is guided by the color image. We set $n = 3$ in all the experiments. The depth refinement algorithm is summarized in Algorithm 1.

Algorithm 1 Overview of the depth refinement procedure.

Input: Color images $I_1 \ldots I_N$, patch size $n\times n$ and depth maps $D_1 \ldots D_N$.
Output: Refined depth map D_1 for color image I_1.

1: Calculate photo-consistency cost $C_{pixel}(p)$ in I_1 and edge preserving cost $C_{patch}(p)$ in I_1.
2: Calculate matching cost $C(p) = C_{pixel}(p) + C_{patch}(p)$.
3: **if** (matching cost $C(p)$ > average cost of patch P $(n\times n)$ centered on p **then**
4: **for** pixel $q_i \in$ patch P **do**
5: Find q_i with the lowest matching cost $C(q_i)$ and replace the depth and matching cost of p with q_i's.
6: Run weighted median filter.

4.2. View selection

A large number of works (e.g., [3–5]) improve the quality of synthesized images by correcting misalignment or filling holes. However, less attention has been paid to select input views, which is also important for improving the quality of synthesized images. Previous studies may choose incorrect or redundant views based on angles or distances between two views, which often leads to blurry synthesized images. In order to avoid choosing such input views, we select views not only considering angles and distances but also overlaps between two views.

Fig. 4 shows the selection process. Firstly, the distance between the input and the target views is calculated as shown in Fig. 4(a), where A, B, C, D, E, F are input views and T is the target view. The distance between the input and target views is defined by:

$$distance(i) = \|O_t - O_i\|, \tag{4}$$

where O_t and O_i are the centers of target view t and input view i, respectively.

Then we rank the calculated distances and select the top ten images as a local group. From these local images, angles between the target and input views are calculated:

$$angle(i) = \arccos\left(\frac{\vec{n_t}\cdot\vec{n_i}}{\|\vec{n_t}\|\cdot\|\vec{n_i}\|}\right), \tag{5}$$

where $\vec{n_t}$ and $\vec{n_i}$ are the view directions of target view t and input view i, respectively.

If the angle is bigger than the field of view of the camera capturing input images, we get rid of it from the local input image group as shown in Fig. 4(b).

Furthermore, in order to remove views, like A which has a small angle and distance, but no overlap with the target view, we calculate the overlap between input and target views. If the overlap is zero, we remove it from the local image group (Fig. 4(c)). To reduce the computation time of calculating overlaps, we downsample the input image with an equal sampling interval and only project sampled pixels into the target view. In this way, the computation time can be reduced depending on the sampling interval. In our experiment, all the depth images have the same world coordinate system.

The target virtual image is synthesized by locally blending the input images. However, directly blending all images is time-consuming. So that, we use a variable number of input images to produce the virtual image. We first project an input image which is selected based on our view selection approach in the input image group to the virtual position, and then detect the holes in the virtual image. If the size of the largest hole is bigger than a threshold (e.g., 0.04% of the whole image), we then choose another input image to fill holes. We iteratively run this process until the largest hole has been sufficiently covered.

4.3. Layered 3D warping

The whole pipeline of our layered 3D warping is described in Fig. 5. The core part of IBR methods is 3D warping. It projects textures and depth information from a reference image plane to new positions in the target image plane using the corresponding camera's intrinsic and extrinsic matrices.

Fig. 6 shows the projection process. Let $\tilde{p_1} = [u, v, 1]^T$ be a pixel point in the image plane C_1. P_1 is projected into the world coordinate system at $\tilde{P} = [X, Y, Z, 1]^T$. The relationship between $\tilde{P}$ and $\tilde{p_1}$ can be defined by left camera's intrinsic matrix K_1, rotation matrix R_1 and translation matrix T_1:

$$\begin{aligned} \tilde{p_1} &\cong M\tilde{P} \\ \tilde{P} &\cong MM^{+} + \tilde{p_1}, \end{aligned} \tag{6}$$

where $M = K_1[R_1|T_1]$. $\cong$ is transformed in metric positions using the Z coordinate position of 3D object point in the camera coordinate system. Furthermore, $\tilde{P}$ is projected into the image plane C_2 at the pixel position $\tilde{p_2} = [u', v', 1]^T$, which is calculated by

$$\tilde{p_2} \cong M' * \tilde{P}, \tag{7}$$

where $M' = K_2[R_2|T_2]$. K_2, R_2 and T_2 represent the intrinsic, rotation and translation matrices of the right camera.

However, the projection errors caused by 3D warping or incorrect depth information often lead to background information wrongly appear in the foreground, as shown in Fig. 7.

To solve this problem, we evenly divide the depth image into layers based on the maximum and minimum depth values. Fig. 8 shows an example of the layered depth maps. On each layer, we apply 3D warping with corresponding color-and-depth image pairs to produce new images and then employ a switching median filter to remove unusual pixels in each new image.

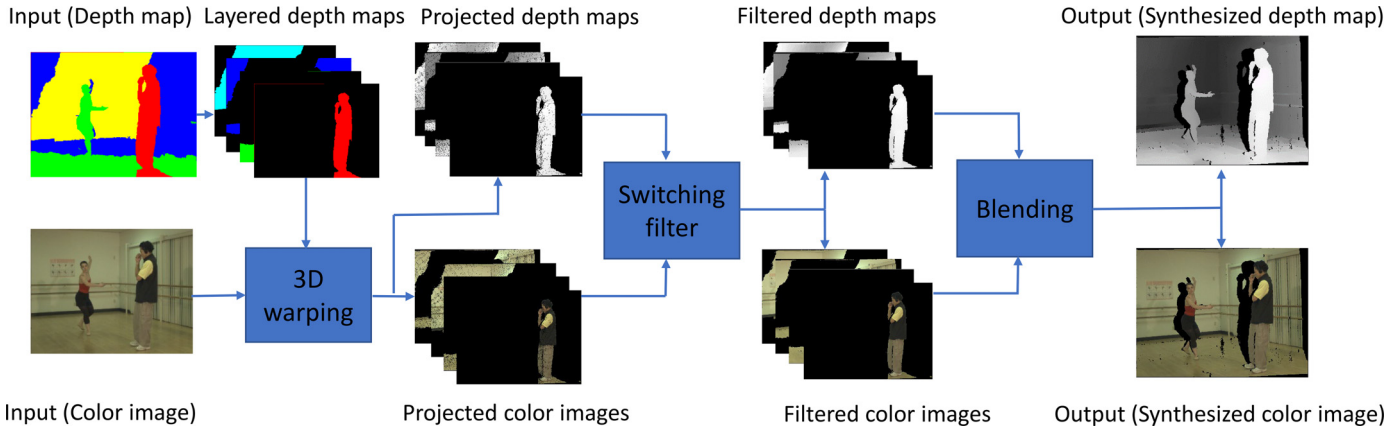


Fig. 5. Layered 3D warping. The input are color-and-depth image pairs. Based on the maximum and minimum depth values, the depth map is divided into layers. On each layer, we apply 3D warping to synthesize the new image. A switching median filter is applied to fill missing information in these images. After that, all the filtered images are blended to produce the color-and-depth image pairs.

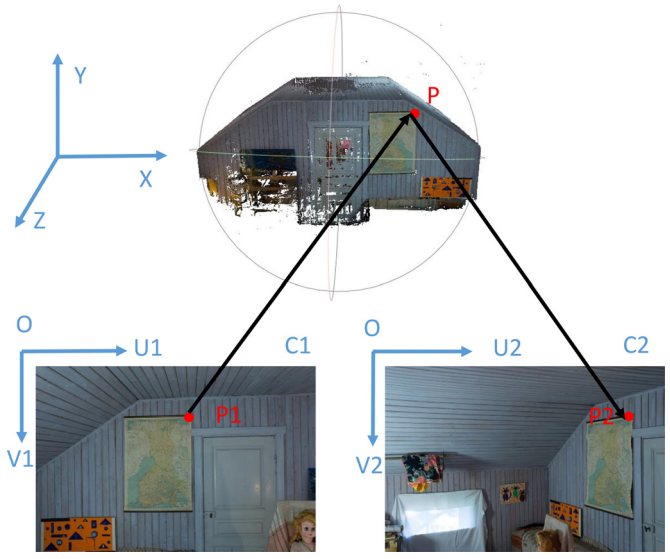


Fig. 6. 3D warping. A point P_1 in the image plane C_1 is projected to a world point P and then P is projected to another image plane C_2 at position P_2.

Fig. 7. Problems caused by 3D warping in the synthesized image.

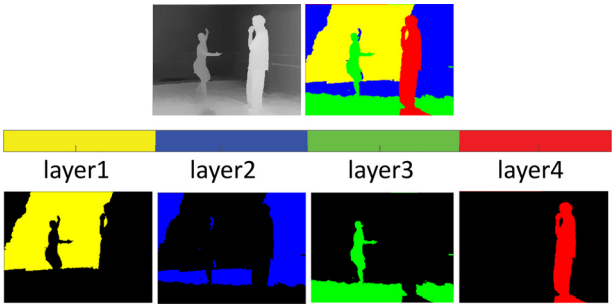


Fig. 8. The layered depth maps. A depth map is evenly divided into layers based on the maximum and minimum depth values.

A median filter [32] is often used to filter unusual pixels in the projected image, for the distribution of these pixels has similar characteristics as salt-and-pepper noise. However, the traditional median filter is implemented uniformly across the whole image and tends to modify both noisy and good pixels at the same time. As a result, the filtered images are more likely to lose some details such as edges and small textures. Unlike the median filter, our switching median filter only refines pixels that have unusual information and is able to avoid smoothing over images. The switching median filter for pixel $P_{i,j}$ is defined as follows:

$$P_{i,j} = \begin{cases} median\left\{ P'_{i+u,j+v} | (u,v) \in W \right\} & \text{if } P_{i,j} \in S \\ P_{i,j} & \text{otherwise} \end{cases}, \tag{8}$$

where $median$ is the traditional median filter and $P'_{i-u,j-v}$ is a pixel in the median kernel, $W = \{(u,v) | -(N+1)/2 \leq (u,v) \leq (N+1)/2\}$, N is the size of the median kernel and S is a cluster of chosen pixels. If $P_{i,j}$ is equal to zero and more than half of the pixels centered on $P_{i,j}$ are non-zeros, we consider $P_{i,j}$ belong to S.

After performing the switching median filter, we blend these new images together to produce the final synthesized images. We found four layers to be a good trade-off between quality and speed.

4.4. Imperfections of IBR and solutions

In this section, we explain the imperfections of IBR and summarize the solution for each of them. There are four basic problems in IBR, that are ghost contours, cracks, occlusion, and holes.

Ghost contour. The ghost contour is mainly caused by the edge misalignment of object boundaries between a color image and its corresponding depth map. Object edges in the color image always contain

Fig. 9. Example results of synthesized images from different scenes (left to right): Office, Dorm, Playroom, Reading corner, Attic.

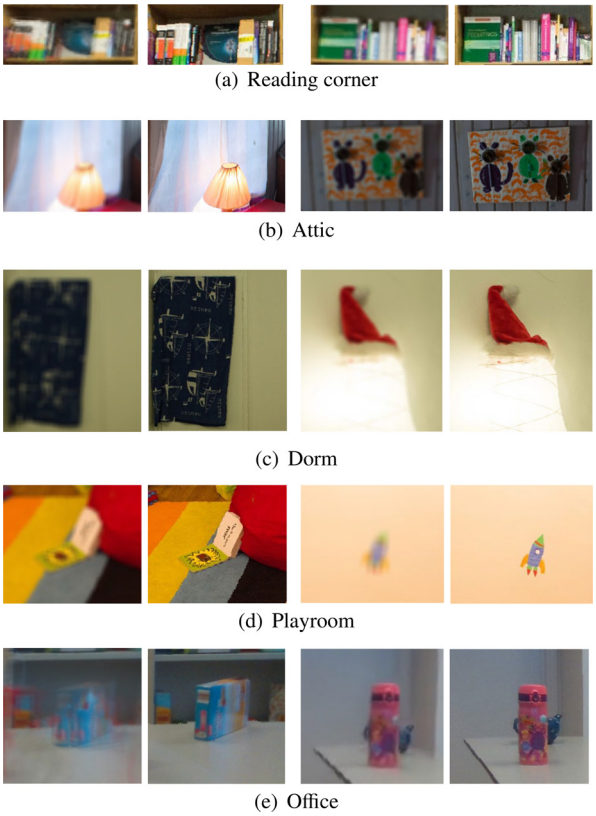


Fig. 10. Qualitative comparison between Local Light Field Fusion [36] (first and third columns) and our approach (second and fourth columns) on five datasets.

Table 1
Quantitative evaluation of rendered depth maps on different datasets.

	Lab	Reading corner	Playroom	Dorm	Breakdancers
RMS	0.401	0.454	0.468	0.499	2.301
log_{10}	0.043	0.058	0.061	0.063	0.137

transitional pixels, while edges in its related depth map do not have such transitional regions. After projection, the transitional areas of the image are split and appear various visual artifacts.

To address this challenge, we refine the initial depth map by correcting the misalignment of boundaries between color-and-depth image pairs and filling missing depth information (see Section 4.1). In addition, when blending projected images, we detect big holes in the projected image and dilate these holes with several pixels, which is also useful to remove ghost contours.

Cracks. Due to the miss-focus and non-integer index problems, the input pixel is usually not projected to a point at an integer position. After resampling, there may be more than one value in a position, while there are no values in other positions, which results in crack artifacts in the projected image.

The median filter is often used to remove cracks. However, traditional median filter often leads to the over-smoothing problem, which makes images lose small details. We introduce the switching median filter that only filters pixels among cracks to avoid above issues (see Section 4.3).

Occlusion. When objects in the background and foreground are projected to the same position, objects in the foreground may be occluded by objects in the background, which is caused by the incorrect depth information. Besides, objects that are supposed to show correctly can also be occluded due to the projection errors. The Z-buffer approach is the most commonly used method to address these problems. However, it is not sufficient to generate high-quality synthesized images, since there are still many background pixels appearing in the foreground after applying this approach, especially for dynamic scenes.

To address this problem, we combine the layered 3D warping and switching median filter to synthesize new images (see Section 4.3). The layered depth map has the ability to represent geometry of occluded elements and the switching median filter can reduce the loss of visible information. These two components are designed to work together, giving high-quality performance.

Holes. Unobserved regions will lead to holes in synthesized images. Moreover, the fixed number of input views used by traditional view synthesis methods is not guaranteed to cover the whole virtual view, which results in holes during rendering.

Unlike previous algorithms using a fixed number of input images, we use an adaptive number of images to fill holes in the synthesized image (see Section 4.2). Our adaptive view selection approach makes

sure the given virtual view can be sufficiently covered, which avoids big holes in the synthesized image. At the same time, our approach is able to limit the input views used for rendering. This helps to avoid blurry synthesized images and improve the rendering speed.

5. Experimental results

We evaluate the proposed approaches on seven static datasets including our three own datasets (Study room, Office and Lab), four datasets (Attic, Dorm, Playroom, Reading corner) from [15], and two dynamic datasets (Ballet and Breakdancers) from [33]. There are less than 220 color-and-depth image pairs in the seven static datasets. The depth images in these datasets are calculated using a unique world coordinate system and the intrinsic and extrinsic parameters are estimated at the same time. Each of the two dynamic datasets contains a sequence of 100 color-and-depth image pairs, captured by eight static cameras which are positioned along an arc at 20-degree intervals. Like the seven static datasets, the depth images in the two dynamic datasets are also calculated by a unique world coordinate system and the intrinsic and extrinsic parameters are also known.

5.1. Overall performance

We randomly choose an image from the initial captured dataset as our ground truth image and then use the other images to synthesize the chosen image. Fig. 1 shows some examples of synthesized images and their corresponding ground truth images. Fig. 9 shows some additional synthesized images. From Fig. 1 and Fig. 9 we can see that the proposed method is able to provide high-quality synthesized images.

We use root mean squared error (RMS) meters (lower is better) and average log10 error (lower is better) [34] to quantitatively evaluate the synthesized depth maps.

$$RMS = \sqrt{\frac{1}{n}\sum_p^n (y_p - \hat{y}_p)^2}. \tag{9}$$

$$log_{10} = \frac{1}{n}\sum_p^n |log_{10}(y_p) - log_{10}(\hat{y}_p)|. \tag{10}$$

We also randomly choose a depth map from the initial captured dataset as our ground truth map and then use other depth maps to synthesize it. The average number of input depth maps we use is four. The experimental results are summarized in Table 1. From Table 1 we can see that our approach achieves better performance on the static datasets. For example, compared with the RMS of Breakdancers dataset (2.301), the RMS of Reading corner dataset is 0.401, which is relatively small error among popular depth evaluation approaches [35].

5.2. Comparison with other methods

In Fig. 10 we compare our method to Local Light Field Fusion (LLFF) [36] which also uses layered depth maps to synthesize images. Since LLFF is designed for static datasets containing large overlaps, we only compare it on our static datasets with small changes. As we can see, LLFF suffers from the same limitation as other deep learning-based view synthesis methods, as images synthesized by LLFF are blurry. In contrast, our approach can provide sharp synthesized images for various scenes.

In order to quantitatively evaluate the synthesized color images, we compare our method with state-of-the-art learning-based algorithms on static datasets including Lab and Study room datasets. Lab is a dense image set and Study room is a sparse image set. The peak signal-to-noise ratio (PSNR) is used to evaluate image quality, where a higher PSNR value means a better image quality. The experimental results are given in Table 2.

As can bee seen from Table 2, compared with learning-based approaches, our method achieves better performance on Study room. The results indicate that our approach is more robust to scenes consisting of sparsely captured images. Besides, our approach is free from training and can provide plausible synthesized images.

Table 2
The PSNR comparison with different algorithms.

Methods	the average PSNR over 100 images (dB)	
	Lab	Study room
SM[20]	21.39	10.76
LLFF[36]	23.18	14.10
NeRF[37]	**37.65**	21.03
Ours	33.10	**27.85**

Table 3
The PSNR comparison with different algorithms.

Methods	the average PSNR over 100 images (dB)	
	Ballet	Breakdancers
VSRS[38]	30.23	31.17
Liu[7]	32.52	33.33
Dai[6]	32.55	31.77
Loghman[39]	30.36	31.64
Ours	**33.40**	**33.59**

Table 4
Rendering speed comparison with different algorithms.

Methods	the average rendering time per frame (second)	
	Ballet	Breakdancers
Liu[7]	0.30	0.31
Li [2]	1.03	1.029
Ours	**0.28**	**0.30**

We also compare our method with state-of-the-art algorithms which are suitable for dynamic datasets. We use two reference images to synthesize a new image on the Ballet and Breakdancers datasets. The comparison results are given in Table 3. We can see that our algorithm performs the best for both datasets.

To evaluate the rendering speed, we compare our method with the state-of-the-art approaches on the two dynamic datasets. Table 4 summarizes the rendering speed comparison results. It can be seen that our approach achieves the best performance.

5.3. Effect of depth refinement

Fig. 11 shows the PSNR comparison with and without our depth refinement method on different datasets. As we can see, our proposed approach consistently improves the PSNR through all the testing frames.

With the refined depth map, the growth of PSNR in the Ballet dataset is the larger. This is because the number of misalignment between the color image and the depth map, and imprecise depth values in the Ballet dataset is more than those in the other datasets. After the depth refinement, these issues are solved, resulting in the increased PSNR.

Fig. 12 (a) visualizes the depth refinement results on several datasets. We can see that our pixel-to-pixel multi-view depth refinement method is able to improve the quality of the depth map by filling missing depth information or refining incorrect depth values. Fig. 12(b) shows the alignment process where a foreground color pixel A in the object boundary is wrongly assigned a background depth value $A1$, and after the depth refinement, this value is replaced with the correct foreground depth value $A2$ in the refined depth map.

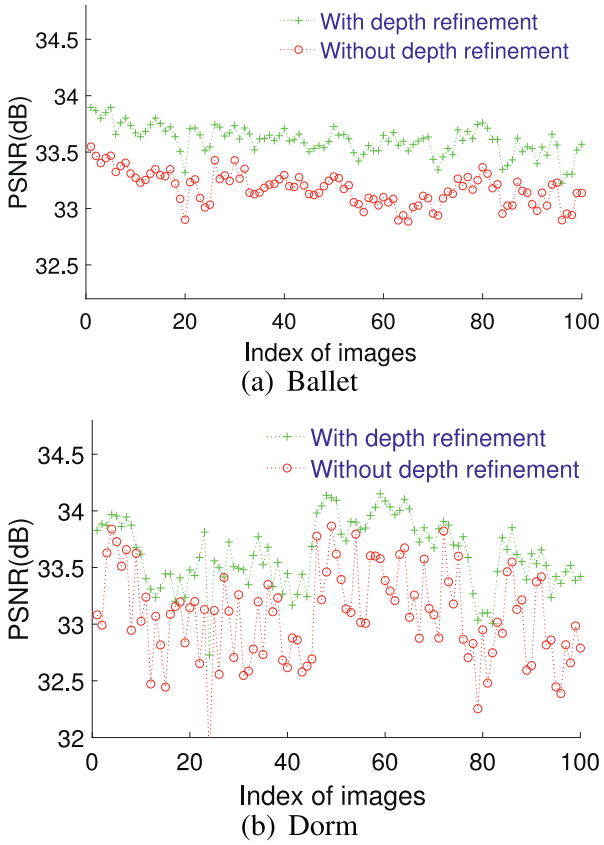


Fig. 11. The PSNR comparison with and without depth refinement on each frame.

Table 5
The PSNR comparison between the guided filter and our depth refinement approach.

	the average PSNR over 100 images (dB)	
	guided filter [40]	ours
Attic	29.39	33.28
Dorm	29.82	33.71
Ballet	28.85	33.43
Office	30.37	34.31
Lab	30.61	33.21
Playroom	29.44	33.53
Study room	27.87	31.75
Breakdancers	30.29	33.89
Reading corner	28.53	31.78

Furthermore, we compare our depth refinement algorithm with the guided filter [40], which is designed to produce edge-preserving depth maps. We use the color image as the guided image and compare the average PSNR over 100 frames, as shown in Table 5. It can be seen that compared with guide filter, our approach achieves better performance in various scenes.

5.4. Effect of view selection

The quality of synthesized images is influenced by the quantity of well-chosen input images. We compare the hole sizes of synthesized image using different numbers of input views in Fig. 13 and Table 6. For traditional methods, the number of input views is fixed, such as two or three, which does not guarantee to cover the whole virtual view. As a result, big holes often appear in the synthesized image. In contrast, our method with a variable number of input images is able to reduce the hole size significantly. The average number of input views we use is

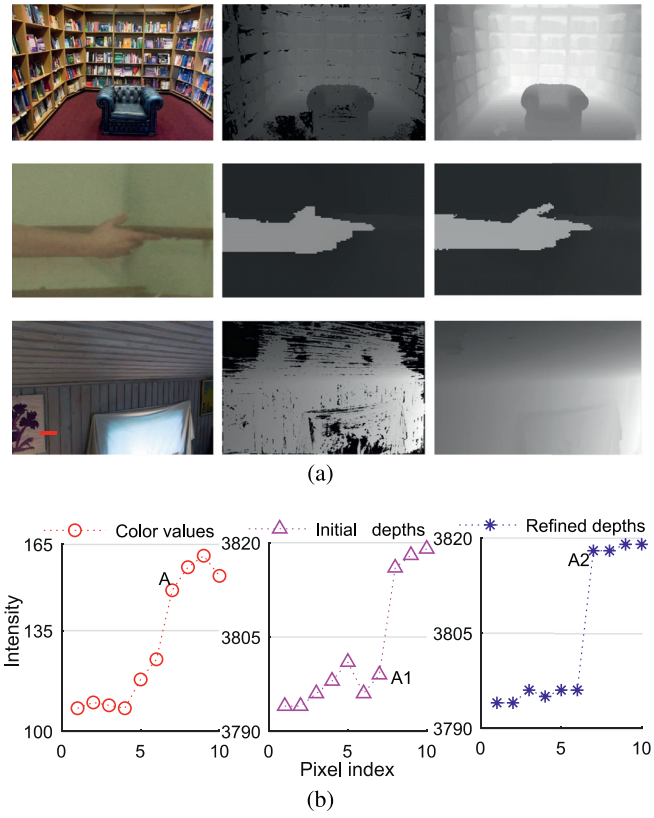


Fig. 12. Example results after depth refinement. (a): The visualization results of depth refinement on different datasets (top to bottom): Reading corner, Ballet and Attic. (b): An example showing the misalignment between the foreground color A and background depth $A1$ is corrected by our depth refinement, where $A1$ is replaced with the correct foreground depth $A2$. The color and depth intensities are obtained along the horizontal red line in the Attic dataset in (a). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Hole size comparison of synthesized images using different numbers of input views. The hole size is defined by the percentage of missing pixels in the whole image.

	Hole size(%)			
	1 view	2 views	3 views	ours (adaptive views)
Attic	50.07	10.31	2.26	0.01
Dorm	60.19	18.35	5.85	0.03
Ballet	50.16	3.15	1.93	0.02
Office	31.13	2.51	1.33	0.01
Lab	35.19	3.24	2.67	0.02
Playroom	45.96	10.51	6.21	0.03
Study room	60.19	20.15	7.82	0.02
Breakdancers	47.89	2.15	1.21	0.01
Reading corner	23.12	10.49	1.18	0.02

four. For example, for the Study room dataset, the hole size is reduced by 20.13%, and 7.80% compared to methods with two and three input views, respectively. This is because the captured images in Study room dataset are very sparse. If the virtual view is substantially different from the input images, the synthesized image produced by the fixed number of input images may still have large holes.

5.5. Effect of layered 3D warping

Fig. 14 shows the PSNR comparison results between layered 3D warping and Z-buffer on two dynamic datasets that are more challenging

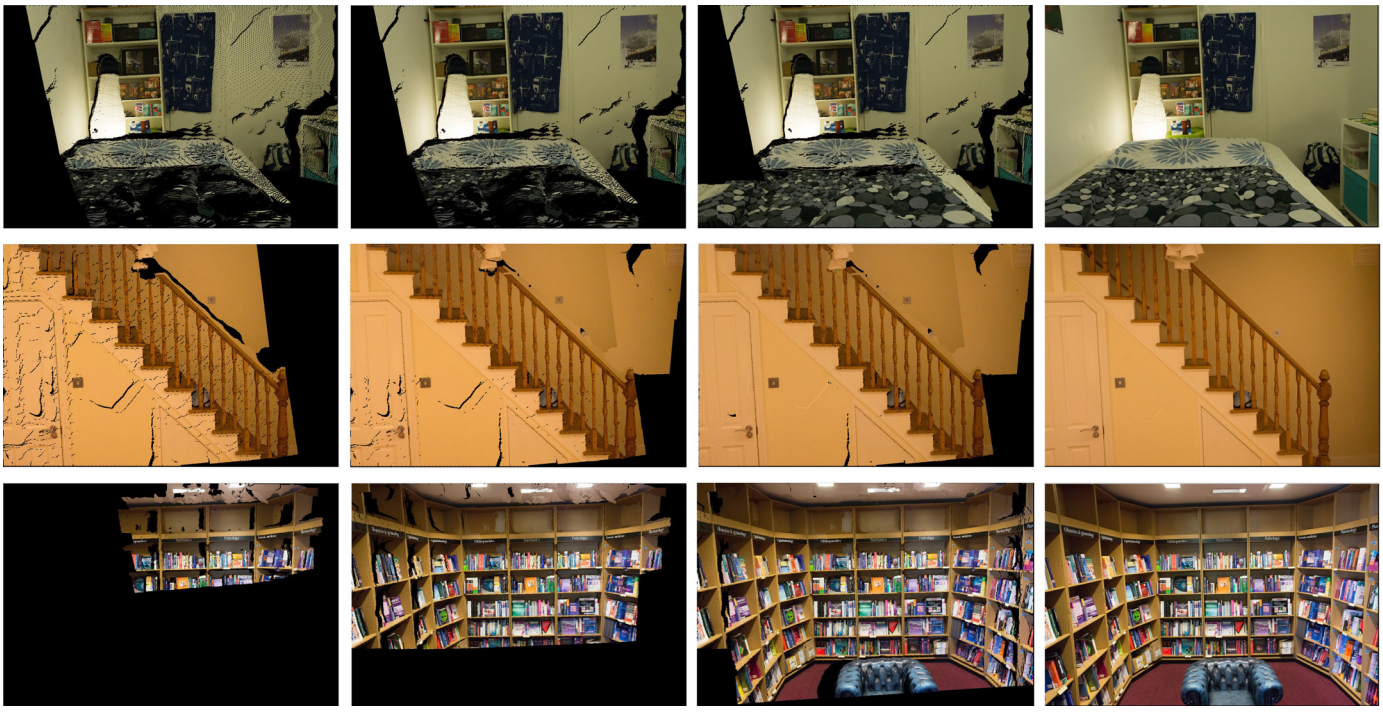
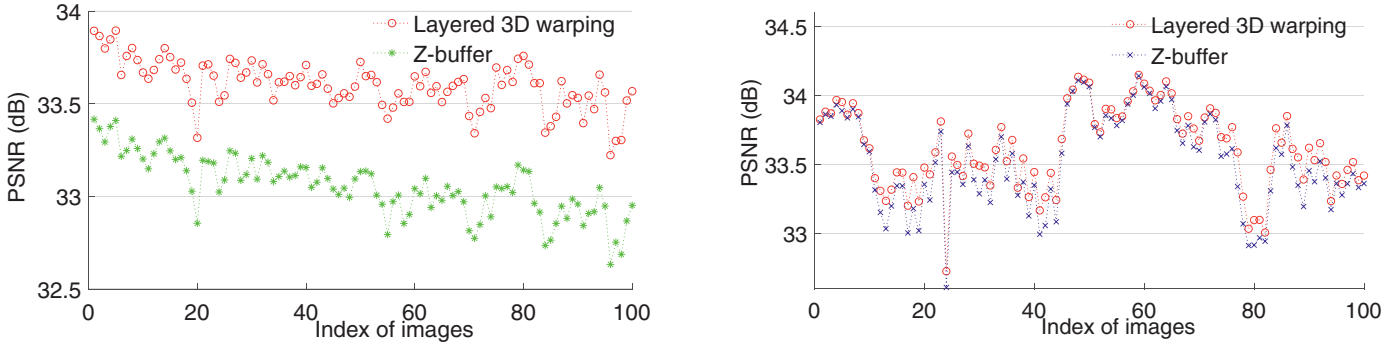


Fig. 13. Synthesized images by one view (first column), two views (second column), three views (third column) and adaptive views (fourth column) on different datasets.



(a) Ballet (b) Breakdancers

Fig. 14. The PSNR comparison between layered 3D warping and Z-buffer on each frame.

than static ones. We can see that our layered 3D warping consistently improves the PSNR through all the testing frames.

The effectiveness is also demonstrated by Fig. 15 which shows some snapshots of synthesized images with our layered 3D warping on different datasets. As can be seen, the background pixels are adequately removed and replaced by correct foreground pixels after the layered 3D warping.

We introduce the switching median filter to fill missing information in layered synthesized images. Compared with the traditional median filter, the main advantage of our switching median filter is to avoid over-smoothing. To verify its effectiveness, we compare the average PSNR over 100 frames with the median filter [32], as shown in Table 7. It can be seen that our approach achieves better performance in different scenes.

Table 7
The PSNR comparison between the median filter and switching median filter on different datasets.

	the average PSNR over 100 images (dB)	
	median filter [32]	ours (switching median filter)
Attic	31.57	33.28
Dorm	30.12	33.71
Ballet	31.15	33.43
Office	30.63	34.31
Lab	31.11	33.21
Playroom	30.64	33.53
Study room	29.57	31.75
Breakdancers	31.91	33.89
Reading corner	29.67	31.78

5.6. Effect of multi-layered depth maps

To verify the necessity of depth processing in the view synthesis framework, in Table 8 we calculate the average PSNR over 100 frames on several datasets. The traditional image based rendering, the method combing image based rendering and depth refinement, and the approach combining IBR and layered 3D warping are referenced as IBR, IBR_DR

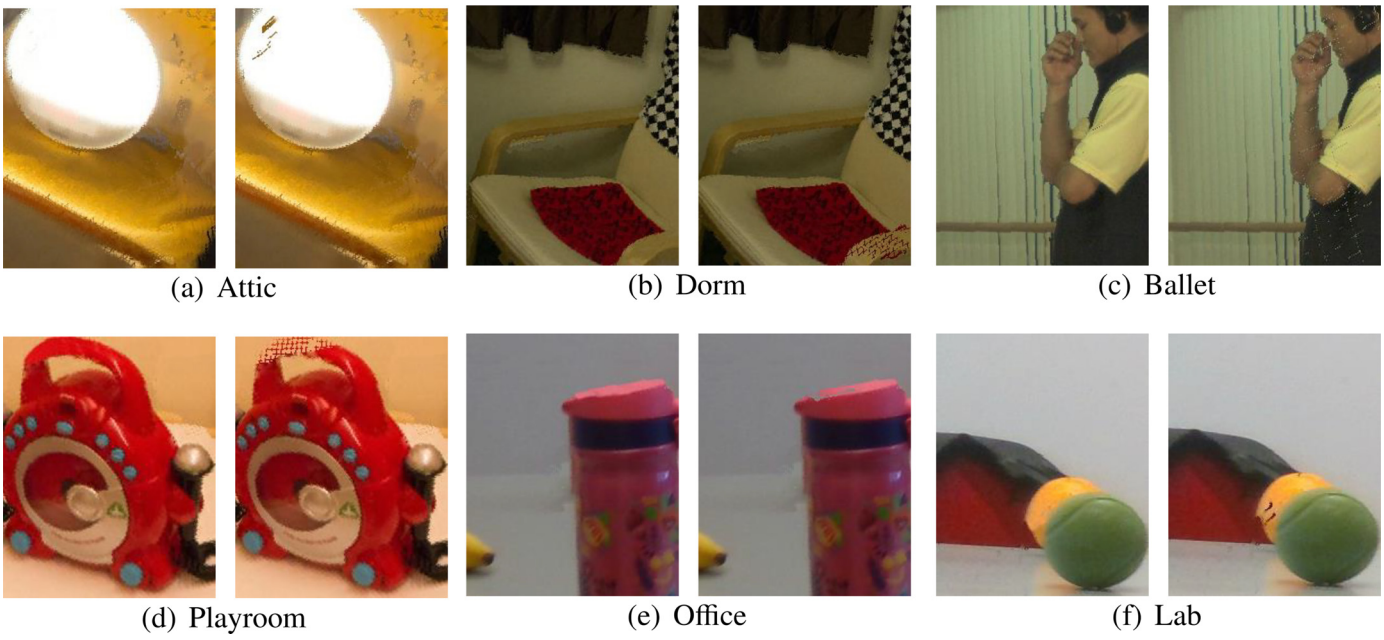


Fig. 15. Synthesized images with (first) and without (second) layered 3D warping on different datasets.

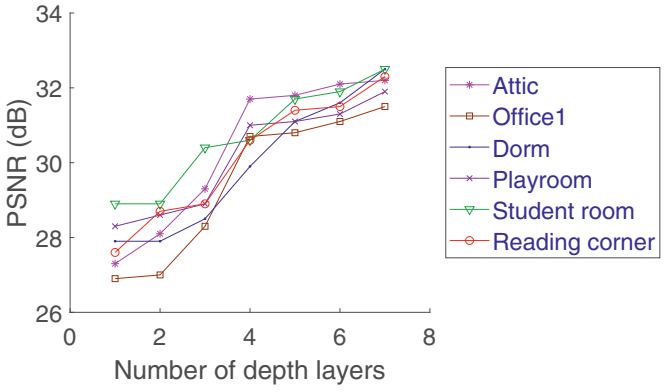


Fig. 16. Performance with different numbers of layered depth maps.

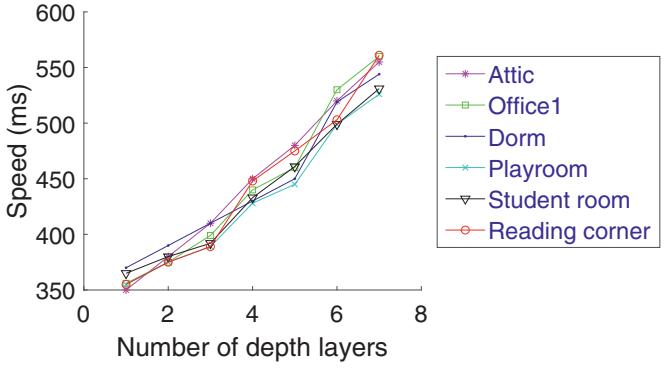


Fig. 17. The rendering speed of our method with different numbers of layered depth maps.

and LIBR respectively. The LIBR_DR is the algorithm combing depth refinement, and layered 3D warping.

From Table 8 we can see that using IBR_DR or LIBR alone improves the performance as they are able to better process depth information, and the combined method LIBR_DR performs best.

Fig. 18. The typical limitations of our approach: artifacts caused by incorrect depth maps.

Table 8
Quantitative evaluation of the synthesized image in terms of PSNR with different approaches.

	the average PSNR over 100 images (dB)			
	IBR	IBR_DR	LIBR	LIBR_DR
Attic	27.99	32.50	31.45	33.28
Dorm	26.89	33.12	31.62	33.71
Ballet	25.43	33.17	33.05	33.49
Office	28.13	33.21	31.33	34.31
Lab	29.19	30.24	30.21	33.21
Playroom	27.96	29.52	30.21	33.53
Study room	28.12	30.22	29.45	31.75
Breakdancers	26.52	33.26	33.53	33.67
Reading corner	23.12	25.16	27.84	31.78

5.7. Quality and time efficiency

The qualitative comparison of synthesized images with different numbers of depth layers is shown in Fig. 16. We can see that more depth layers will improve the PSNR of synthesized images. However, more layers will also increase the computation time, as shown in Fig. 17. We found four layers to be a good trade-off between quality and speed.

6. Conclusion

In this paper we have proposed a novel view synthesis framework that first refines depth maps by correcting misalignment of ob-

ject boundaries between color-and-depth image pairs and filling missing depth information. We then divide the depth map into layers and introduce a fast rendering algorithm combining an adaptive view selection approach and a layered 3D warping to synthesize high-quality free-viewpoint virtual images. The experimental results demonstrate that the quality of synthesized images is improved significantly with refined and layered depth maps. Since the rendering time of our proposed algorithm only depends on the display resolution of synthesized images, it can be used in various applications such as mobile phones and virtual environment [41]. However, some limitations are worth noting. If the object has a large range, it will be divided into different layers, which could be further optimized. Besides, when the depth map produced by the depth camera has too much missing information, the synthesized image generated by our method shows various artifacts, as shown in Fig. 18. Therefore, new methods are required to generate high-quality depth images for scenes with texture-less or reflective objects.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Honglin Yuan: Conceptualization, Methodology, Data curation, Writing - original draft, Writing - review & editing. **Remco C. Veltkamp:** Supervision, Writing - review & editing.

Acknowledgements

Part of the research has been supported by the research start-up funding for talent programs from Nanjing University of Information Science and Technology.

References

[1] Fehn C. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3d-tv. In: Stereoscopic Displays and Virtual Reality Systems XI, 5291. International Society for Optics and Photonics; 2004. p. 93–105.
[2] Li S, Zhu C, Sun M-T. Hole filling with multiple reference views in DIBR view synthesis. IEEE Trans Multimedia 2018;20(8):1948–59.
[3] Mori Y, Fukushima N, Yendo T, Fujii T, Tanimoto M. View generation with 3D warping using depth information for FTV. Signal Process Image Commun 2009;24(1–2):65–72.
[4] Chaurasia G, Duchene S, Sorkine-Hornung O, Drettakis G. Depth synthesis and local warps for plausible image-based navigation. ACM Transactions on Graphics (TOG) 2013;32(3):30.
[5] Lei J, Zhang C, Wu M, You L, Fan K, Hou C. A divide-and-conquer hole-filling method for handling disocclusion in single-view rendering. Multimed Tools Appl 2017;76(6):7661–76.
[6] Dai J, Nguyen T. View synthesis with hierarchical clustering based occlusion filling. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE; 2017. p. 1387–91.
[7] Liu J, Li C, Fan X, Wang Z, Shi M, Yang J. View synthesis with 3D object segmentation-based asynchronous blending and boundary misalignment rectification. Vis Comput 2016;32(6–8):989–99.
[8] Zinger S, Do L, de With P. Free-viewpoint depth image based rendering. J Vis Commun Image Represent 2010;21(5–6):533–41.
[9] Shum H-Y, Chan S-C, Kang SB. Image-based rendering. Springer Science & Business Media; 2008.
[10] Zitnick CL, Kang SB. Stereo for image-based rendering using image over-segmentation. Int J Comput Vis 2007;75(1):49–65.
[11] Tereshin A, Adzhiev V, Fryazinov O, Marrington-Reeve F, Pasko AA. Automatically controlled morphing of 2d shapes with textures. SIAM J Imaging Sci 2020;13(1):78–107.
[12] Wang S, Wang R. Robust view synthesis in wide-baseline complex geometric environments. In: ICASSP. IEEE; 2019. p. 2297–301.
[13] Buyssens P, Daisy M, Tschumperlé D, Lézoray O. Superpixel-based depth map inpainting for RGB-D view synthesis. In: ICIP. IEEE; 2015. p. 4332–6.
[14] Ortiz-Cayon R, Djelouah A, Drettakis G. A Bayesian approach for selective image-based rendering using superpixels. In: International Conference on 3D Vision-3DV; 2015. p. 469–77.
[15] Hedman P, Ritschel T, Drettakis G, Brostow G. Scalable inside-out image-based rendering. ACM Transactions on Graphics (TOG) 2016;35(6):231.
[16] Penner E, Zhang L. Soft 3d reconstruction for view synthesis. ACM Trans Graph 2017;36(6):235:1–235:11.
[17] Flynn J, Neulander I, Philbin J, Snavely N. Deepstereo: Learning to predict new views from the world's imagery. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 5515–24.
[18] Xia F, Zamir AR, He Z, Sax A, Malik J, Savarese S. Gibson env: Real-world perception for embodied agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 9068–79.
[19] Mildenhall B, Srinivasan PP, Ortiz-Cayon R, Kalantari NK, Ramamoorthi R, Ng R, et al. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) 2019.
[20] Zhou T, Tucker R, Flynn J, Fyffe G, Snavely N. Stereo magnification: Learning view synthesis using multiplane images. In: SIGGRAPH; 2018. p. 65:1–65:12.
[21] Srinivasan PP, Wang T, Sreelal A, Ramamoorthi R, Ng R. Learning to synthesize a 4D RGBD light field from a single image. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 2243–51.
[22] Flynn J, Broxton M, Debevec PE, DuVall M, Fyffe G, Overbeck RS, et al. Deepview: View synthesis with learned gradient descent. In: CVPR. Computer Vision Foundation / IEEE; 2019. p. 2367–76.
[23] Ni L, Jiang H, Cai J, Zheng J, Li H, Liu X. Unsupervised dense light field reconstruction with occlusion awareness. Comput Graph Forum 2019;38(7):425–36.
[24] Schmeing M, Jiang X. Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting. IEEE Trans Multimedia 2015;17(12):2160–73.
[25] Bertalmio M, Sapiro G, Caselles V, Ballester C. Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co.; 2000. p. 417–24.
[26] Solh M, AlRegib G. Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video. IEEE J Sel Top Signal Process 2012;6(5):495–504.
[27] Chen W-Y, Chang Y-L, Lin S-F, Ding L-F, Chen L-G. Efficient depth image based rendering with edge dependent depth filter and interpolation. In: 2005 IEEE International Conference on Multimedia and Expo. IEEE; 2005. p. 1314–17.
[28] Sinha SN, Steedly D, Szeliski R. Piecewise planar stereo for image-based rendering. In: ICCV. IEEE Computer Society; 2009. p. 1881–8.
[29] Goesele M, Ackermann J, Fuhrmann S, Haubold C, Klowsky R, Steedly D, et al. Ambient point clouds for view interpolation. In: ACM Transactions on Graphics (TOG), 29. ACM; 2010. p. 95.
[30] Bleyer M, Rhemann C, Rother C. Patchmatch stereo-stereo matching with slanted support windows.. In: Bmvc, 11; 2011. p. 1–11.
[31] Ma Z, He K, Wei Y, Sun J, Wu E. Constant time weighted median filtering for stereo matching and beyond. In: Proceedings of the IEEE International Conference on Computer Vision; 2013. p. 49–56.
[32] Arias-Castro E, Donoho DL, et al. Does median filtering truly preserve edges better than linear filtering? The Annals of Statistics 2009;37(3):1172–206.
[33] Zitnick CL, Kang SB, Uyttendaele M, Winder SAJ, Szeliski R. High-quality video view interpolation using a layered representation. ACM Trans Graph 2004;23(3):600–8.
[34] Alhashim I, Wonka P. High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:181211941 2018.
[35] Alhashim I, Wonka P. High quality monocular depth estimation via transfer learning. CoRR 2018;abs/1812.11941.
[36] Mildenhall B, Srinivasan PP, Ortiz-Cayon R, Kalantari NK, Ramamoorthi R, Ng R, et al. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) 2019;38(4):1–14.
[37] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: representing scenes as neural radiance fields for view synthesis. arXiv preprint arXiv:200308934 2020.
[38] Software for view synthesis. Software for view synthesis. http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/mpeg2/VS.htm; 2020.
[39] Loghman M, Kim J. Segmentation-based view synthesis for multi-view video plus depth. Multimed Tools Appl 2015;74(5):1611–25.
[40] He K, Sun J, Tang X. Guided image filtering. IEEE Trans Pattern Anal Mach Intell 2012;35(6):1397–409.
[41] Yuan H, Veltkamp RC. Presim: a 3d photo-realistic environment simulator for visual AI. IEEE Robotics Autom Lett 2021;6(2):2501–8.