Computational Visual Media
https://doi.org/10.1007/s41095-022-0323-3 Vol. 10, No. 2, April 2024, 331–353

Research Article

Real-time distance field acceleration based free-viewpoint video synthesis for large sports fields

Yanran Dai[1], Jing Li[1] (✉), Yuqi Jiang[1], Haidong Qin[2], Bang Liang[2], Shikuan Hong[1], Haozhe Pan[1], and Tao Yang[2]

Abstract Free-viewpoint video allows the user to view objects from any virtual perspective, creating an immersive visual experience. This technology enhances the interactivity and freedom of multimedia performances. However, many free-viewpoint video synthesis methods hardly satisfy the requirement to work in real time with high precision, particularly for sports fields having large areas and numerous moving objects. To address these issues, we propose a free-viewpoint video synthesis method based on distance field acceleration. The central idea is to fuse multi-view distance field information and use it to adjust the search step size adaptively. Adaptive step size search is used in two ways: for fast estimation of multi-object three-dimensional surfaces, and synthetic view rendering based on global occlusion judgement. We have implemented our ideas using parallel computing for interactive display, using CUDA and OpenGL frameworks, and have used real-world and simulated experimental datasets for evaluation. The results show that the proposed method can render free-viewpoint videos with multiple objects on large sports fields at 25 fps. Furthermore, the visual quality of our synthetic novel viewpoint images exceeds that of state-of-the-art neural-rendering-based methods.

1 School of Telecommunications Engineering, Xidian University, Xi'an 710071, China. E-mail: Y. Dai, yrdai@stu.xidian.edu.cn; J. Li, jinglixd@mail.xidian.edu.cn (✉); Y. Jiang, imjiangyq@stu.xidian.edu.cn; S. Hong, sk_hong@stu.xidian.edu.cn; H. Pan, haozhepan@stu.xidian.edu.cn.

2 National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, SAIIP, the School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China. E-mail: H. Qin, qhd@mail.nwpu.edu.cn; B. Liang, knl000b@mail.nwpu.edu.cn; T. Yang, tyang@nwpu.edu.cn.

Manuscript received: 2022-04-13; accepted: 2022-10-26

Keywords free-viewpoint video; view synthesis; camera array; distance field; sports video

1 Introduction

Free-viewpoint video techniques synthesize arbitrary virtual viewpoint images by fusing image information from multiple views. Free-viewpoint video can provide a 6-DoF (degrees of freedom) viewing experience, which breaks the limitation of traditional video that can only display fixed two-dimensional visual content. Traditional video presentation depends on the position and angle of the camera, so content can only be passively switched between a few discrete viewpoints. However, the free-viewpoint video allows users to interactively set up virtual camera trails. The virtual camera can be set not only at the reference viewpoints, but we can even place virtual camera trails in locations where they cannot be physically installed, such as flying over or walking through. Free-viewpoint video brings a novel and immersive visual experience to TV broadcasting of sports events, galas, variety videos, etc.

Free-viewpoint video synthesis still faces many challenges in practical applications, especially on large sports fields. On the one hand, the lighting conditions and even camera installation positions are limited. On the other hand, factors such as broad coverage, sparse viewpoints, multiple targets, and complex occlusion relationships also affect the quality and efficiency of free-viewpoint video synthesis. Many scholars have conducted intensive research to obtain high imaging performance, real-time rendering, and clean algorithmic architectures. Moreover, this research is not limited to academia, and companies such as Intel, Canon, and Sony have also worked on

free-viewpoint video applications.

The aim of our research was to improve the efficiency of free-viewpoint video generation for large scenes without loss of visual quality. We thus explored the factors that affect synthesis efficiency. First, reconstructing the entire 3D structure from multiple views is time-consuming. Restoring only the 3D information and textures needed for the virtual viewpoint can radically reduce the computation. Second, much time is spent searching for intersections of rays with 3D surfaces. Especially for large scenes, the small volume of the objects and their discrete distribution results in many searches without an intersection. If the search step size can be changed intelligently to cross empty regions, new view synthesis efficiency will be significantly improved. In addition, a suitable parallel computing architecture will further enhance the execution speed of the algorithm.

Based on the analysis above, this paper proposes a real-time free-viewpoint video synthesis method accelerated by use of distance fields. The proposed algorithm has two main components: distance field guided variable-step object 3D surface accelerated search and global occlusion judgement. We only restore 3D surfaces and texture as seen from the synthesis viewpoint. Therefore, the virtual viewpoint emits rays at the imaging resolution. Multi-view silhouette maps are used as constraints to search for their intersections with the objects. Each search ray is first limited by a bounding box to narrow the search range. Then, the variable step size guided by the distance field is utilized to accelerate the ray search. Completing all ray searches gives the 3D surfaces from the virtual viewpoint. Furthermore, the variable-step search method can also be used for global occlusion judgement in the rendering stage. To ensure the realism of the synthesized new viewpoint graphics, we choose the unoccluded reference viewpoint closest to the virtual view to provide texture information. The rays are emitted from the 3D surfaces towards the reference viewpoints in order of similarity. We can quickly perform occlusion judgements based on variable step size guided by distance field in order of similarity. To further accelerate free-viewpoint video synthesis for large scenes, we implement our approach in parallel, with interactive control, based on the CUDA and OpenGL frameworks.

The main contributions of this paper are threefold:

- a variable-step search method guided by distance fields, which can cross empty volumes to quickly obtain multi-object 3D surface information,
- a global occlusion judgement method guided by distance fields, which selects the unoccluded reference viewpoint most similar to the virtual view to provide texture information, and
- an experimental, interactively controlled, parallel implementation using CUDA and OpenGL for algorithm evaluation.

2 Related work

Free-viewpoint video synthesis is a long-standing and well-explored problem, with much excellent work published in the last decade. Free-viewpoint video synthesis methods can be divided into three categories: image-based rendering, image-based modeling, and neural volume rendering.

2.1 Image-based rendering

Image-based rendering directly generates images from new viewpoints through image transformation and fusion according to the geometric and depth information from the multiple views. Traditional image rendering methods not only have many limitations in terms of camera layout, but also require a large amount of raw data [1–4]. For example, Fukushima et al. [1] proposed a free view image rendering method using a disparity optimization method called multi-pass dynamic programming (MPDP), achieving real-time performance at 384 × 288 resolution. However, its cameras must be evenly arranged around a rectangle, and it still requires significant computational resources to obtain geometric information from multiple images. Simplifying the geometric information required for viewpoint transformation can improve the efficiency of new viewpoint synthesis. KDDI researchers proposed a series of billboard-based free-viewpoint video synthesis methods [5–11]. These methods first model the foreground in a multi-viewpoint image as a billboard, and then use geometric transformation and inter-frame interpolation to approximate the content. The content to be render from a virtual viewpoint is placed in a reasonable 3D position. These methods have a simple structure and require a small amount of computation, so are widely used for new viewpoint

synthesis in sports. However, they also have obvious drawbacks. When objects occlude each other, they are represented as a single billboard model. Multiple objects are geometrically transformed as a whole, making the billboard model visually unrealistic.

Depth-image-based rendering techniques can additionally obtain depth information for the target surface directly from the depth sensor, with advantages for efficiency and quality of image rendering [12–19]. However, synthesis quality is affected by inaccurate depth information, which can cause voids, cracks, distortions, ghosting, etc. In 2020, Carballeira et al. [12] used 9 stereo-based depth cameras to obtain color and depth information and synthesized a new view in real time through a 3D image warping equation. The method utilizes a layered approach, combining multiple reference camera background and foreground layers to obtain a high-quality synthetic image. Depth-image-based rendering (DIBR) techniques have many advantages but are susceptible to sensor performance and accuracy issues. Benefiting from the development of deep learning, many scholars have proposed novel viewpoint synthesis technology based on neural networks [20–28]. In 2018, Zhou et al. [21] designed a depth network to predict multi-plane images from input stereo images. The network can synthesize a range of new views, including views that significantly exceed the input baseline. In 2019, Flynn et al. [25] incorporated occlusion reasoning and proposed a new method of view synthesis using multiple planes. This approach requires only sparse input view images and performs well for challenging scenes. Although deep learning methods provide novel ideas for new viewpoint synthesis, the generalization ability of their trained models still needs to be improved.

2.2 Image-based modeling

Image-based modeling uses multiple images to recover three-dimensional information about objects [29–35]. The visual hull reconstruction method obtains multiple object hulls from multiple images. The method quantizes three-dimensional space into voxels and then sculpts the space using multi-view silhouettes. The algorithm has low complexity and strong calibration robustness, and is not limited by the camera baseline or application environment. Therefore, it is often used in free-viewpoint video production. For example, Nonaka et al. [36] calculated the silhouette of an object on planes in virtual space by simple projection from video images to 3D space, and expressed the overall shape of the object by combining the planes. The method can obtain contour intersections conveniently and quickly, and can be executed in parallel on GPU hardware. In 2018, Yusuke et al. [37] proposed a 3D voxel model reconstruction method combining object contours and image consistency. Compared to the average color method, the image quality of this method is improved. However, image quality is greatly reduced when the voxels are similar in color or located at an edge. In 2019, Chen et al. [38] proposed a parallel free-viewpoint video synthesis method based on the visual hull and applied it to volleyball and judo. This algorithm first reconstructs a sparse point cloud using a volumetric visual hull and labels 3D ROIs for each object. Next, the reconstruction of dense point clouds is carried out only in the ROI. The reconstructed appearance renders non-occluded and occluded regions with the nearest camera and its neighbors, respectively. Although the present visual hull method has excellent performance [36–39], it is still difficult to provide high voxel resolution results in real time.

2.3 Neural volume rendering

Neural rendering methods generate images or video by tracing a ray into the scene and taking an integral of some sort over the length of the ray [40]. Recently, the outstanding effectiveness of the neural radiance and density field (NeRF) [41] approach has attracted the attention of a large number of researchers; its core idea is to approximate a continuous scene as a neural network, in which the weights store the volumetric scene representation. This network is trained on images captured from multiple views. After training is complete, a clear scene picture can be rendered from the given viewpoint. Scholars have considered how to improve the performance of NeRF in terms of enhancing the training and rendering speed, as well as how to reduce the number of training views, and how to adapt it to different applications. In 2021, Lombardi et al. [42] presented the mixture of volumetric primitives (MVP) approach that combines the completeness of volumetric representations with the efficiency of primitive-based rendering. They used approximately 100 synchronized color cameras arranged in a spherical pattern for data acquisition.

The experimental results demonstrate that this method can generate higher-quality, drivable 3D models quickly. In the same year, Yu et al. [43] accelerated NeRF-based rendering by 5 orders of magnitude using the Plenoctree, which stores the appearance and density values required to model the radiation at a point in the volume. In 2022, Fridovich-Keil et al. [44] published a new study based on Plenoxels. Unlike Plenoctrees, Plenoxels is intended to improve the training efficiency of NeRF. The Plenoxels provides an explicit volumetric representation, based on a view-dependent sparse voxel grid without any neural networks. Plenoxels method is two orders of magnitude faster than neural radiance fields with no loss of visual quality. In 2022, researchers of NVIDIA [45] published work that was able to train high-quality neural network primitives in seconds and render them at 1920×1080 resolution in tens of milliseconds. Their design of a multi-resolution hierarchy of hash tables reduces training and evaluation costs. To synthesize novel views from a set of sparse views, Peng et al. [46] proposed the Neural Body, a new human body representation. This method assumes that the learned neural representations for different frames share the same set of latent codes anchored to a deformable mesh. Observations across frames can be naturally integrated to enrich the training data. To address the problem of lack of input views, DS-NeRF [47] was proposed by Deng et al. They used the sparse point cloud generated by SFM as free deep supervision during training to normalize the learning geometry. Martin-Brualla et al. [48] proposed a system called NeRF-W, which introduces a series of extensions to NeRF to enable accurate reconstruction from unstructured image collections. These excellent studies based on neural voxel rendering work well on object information-rich datasets. However, in a large sports scene, the athletes that viewers focus on are often a small part of the picture. Discrete arrays of synchronized cameras are used to capture dynamic video. Both the camera angle and the number of cameras are limited by the venue in which they are set up. All of these limitations pose a significant challenge to existing NeRF methods.

In large environments, it is difficult for new viewpoint synthesis methods to meet the requirements of synthesis accuracy and efficiency. Synthetic image quality is closely related to the number and resolution of sampled viewpoints, but the more sampled viewpoints and the higher the sampling resolution, the greater the computation. That means new viewpoint synthesis methods for large scenes must overcome the problems of sparse sampling viewpoints, broad coverage, and multiple objects. In large scenes, objects have few features and vary greatly in depth, which can cause image-based rendering methods to fail. Model-based methods are computationally inefficient due to the large amount of uninteresting space contained in the coverage area. Neural voxel rendering-based methods perform poorly given sparse training viewpoints and have limited scene generalization ability. We consider object visualization from virtual viewpoints for depth estimation and rendering, and propose a variable-step guided by distance fields to accelerate the search process.

3 Method

3.1 Overview

We now detail our free-viewpoint video synthesis method based on distance field acceleration. Figure 1 shows the overall architecture of the algorithm. It has three components: data preparation, distance field guided variable-step object 3D surface accelerated search, and global occlusion judgement. Prepared data include multi-view images, silhouette images, distance fields, and camera calibration parameters. Our method integrates this multi-view information and utilizes it to improve the efficiency of acquiring multiple object 3D surfaces and rendering new viewpoints. Only recovering geometric and texture information seen from the virtual viewpoint can fundamentally reduce the computation. Therefore, we initialize a cluster of search rays from the imaging plane of the virtual viewpoint. Then, as shown in Fig. 1(bottom, left), we traverse from the starting point with a variable step size guided by the distance field. The traversed 3D points are projected onto silhouette images of different reference views to determine whether they belong to the 3D surfaces. When all ray searches are complete, we have obtained the 3D surfaces from this novel viewpoint. To obtain the texture of these 3D surfaces, we adopt the global occlusion judgement method based on distance field feedback, as shown in Fig. 1(bottom, right). We

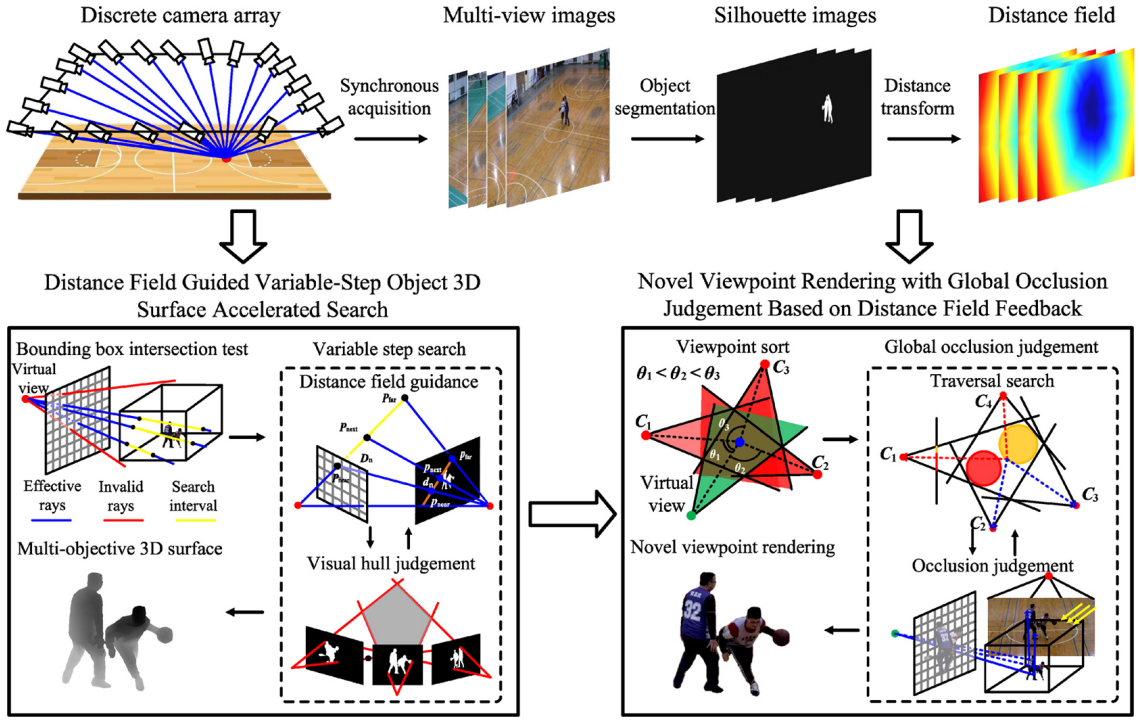

Fig. 1 Overview of the proposed real-time free-viewpoint video synthesis method based on distance field acceleration. Above: data preprocessing of multi-view images, silhouette images, and distance fields. Below: proposed method, with two main parts: distance field guided variable-step object 3D surface accelerated search and global occlusion judgement. Light emitted from the virtual viewpoint is determined by bounding box intersection tests and variable-step search to obtain a multi-objective 3D surface. Based on the 3D surface information, the unobstructed reference viewpoint that most closely resembles the virtual viewpoint is selected to render the novel viewpoint in color.

judge occlusion from all viewpoints for each point on the 3D surfaces, allowing the best unoccluded view to be selected.

3.2 Data preparation

Our novel free-viewpoint video synthesis method needs to prepare the following data: multi-view images, silhouette images, distance fields, and multi-camera internal and external calibration parameters. These data are the basis for our algorithm. There are many excellent methods available to obtain these data. In this paper, we do our best to ensure accuracy and efficiency when preparing the data, but this is not our focus. To describe the algorithm process more clearly, we take a basketball scene as a concrete example in this section.

Specifically, we set up a virtual center point, as shown by the red dot in Fig. 1(top, left). Discrete cameras are placed evenly around the basketball court at equal spatial angles. Multiple synchronized videos are captured by the discrete camera array. Once the camera placement of the discrete array has been determined, the internal and external parameters are also unique and determined. Direct linear transformation (DLT) [49] is used to calculate internal and external parameters of these cameras. 3D coordinates of landmarks in the scene and their pixel coordinates the multi-view images are measured accurately in advance. The mapping matrix $\boldsymbol{M}$ is found using these corresponding points. As a result, a 3D search point $\boldsymbol{P} = [x, y, z, 1]^{\mathrm{T}}$ can be mapped to the point $\boldsymbol{p} = [u, v, 1]^{\mathrm{T}}$ in the image coordinate system. The mapping relationship can be expressed as

$$s\boldsymbol{p} = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}|-\boldsymbol{C}]\boldsymbol{P} = \boldsymbol{M}\boldsymbol{P} \tag{1}$$

After estimating the mapping matrix $\boldsymbol{M}$, we can decompose it to extract the intrinsic parameters ($\boldsymbol{K}$) and extrinsic parameters ($\boldsymbol{R}$, $\boldsymbol{C}$). In this context, $\boldsymbol{R}$ represents the camera's orientation in the world coordinate system, $\boldsymbol{C}$ represents the camera's position relative to the world coordinate system, and $\boldsymbol{I}$ represents the identity matrix. In addition to camera parameters, we also need the silhouette images. Various semantic segmentation networks can be used for body segmentation of basketball players. Several state-of-the-art algorithms were tested on a self-built athlete dataset. Based on robustness and accuracy considerations, DeepLabv3+ was selected

to acquire silhouette images in this paper. The DeepLabv3+ [50] model is a pixel-level semantic segmentation model proposed by Google in 2018. It combines the advantages of a spatial pyramid pooling module and encoder–decoder structure, fully extracts multi-scale contextual information from the image, and uses a decoding module to reconstruct an accurate object boundary. In an open large scene, the number of pixels belonging to each object is small, occlusion relationships are complex, and illumination is uneven. These factors can lead to inaccurate segmentation results or even failure. To improve the completeness and overlap rate of the segmentation results, we trained the segmentation network on our self-built athlete dataset.

The process for obtaining the preparatory data is shown in Fig. 1. The multi-view images captured by the discrete camera array are first passed through a semantic segmentation network to obtain silhouette images. Object silhouettes divide pixels into background and foreground sets. These silhouette images are then transformed into a distance field through distance transformation, which records the shortest distance from each point outside the foreground contour to the foreground boundary, using the Euclidean distance metric.

3.3 Search

3.3.1 Concept

Search range and iteration step size determine the accuracy and efficiency of recovering multiple target geometric structures. Especially in relatively large application scenarios such as basketball courts, soccer fields, and stages, the larger the search range is, the more time-consuming it is to obtain the multi-object 3D surface. A fine search step can create more detailed 3D surfaces, but increases the computational cost. Coarse search steps may lead to inaccurate depth estimation, which then affects the final texture rendering. Using a variable-step search method based on distance field guidance can limit the search range and adaptively adjust the search step according to distance from objects. This variable-step search strategy can improve the algorithm efficiency by reducing the number of iteration steps.

3.3.2 Search range narrowed by bounding box

The size of the 3D space for ray search can be obtained in advance. On the one hand, the hotspot of object activity is the space we need to focus on. On the other hand, the public field of view of the discrete camera array limits the coverage. In a basketball scene, the sampling space is usually set to full or half-court size. The region is bounded by a rectangular box containing all objects. Depending on the application scenario and camera array arrangement, we can get the corresponding bounding box to initially filter out redundant computations.

The emitted rays fall into two cases according to whether they intersect the bounding box, as shown in Fig. 2(a). Only rays that intersect the bounding box are effective. An intersection test can eliminate those rays that do not contain information about the search space. In addition, the bounding box also limits the search interval for each search ray. Because of the different incidence angles, each ray has different intersections with the bounding box. Effective rays have two intersection points which are in front and behind the bounding box. The search range is between the two intersection points, as shown in Fig. 2(b).

3.3.3 Variable search step guided by distance field

The bounding box eliminates ineffective rays and shortens the search range. We next give a method for accelerating the acquisition of 3D surface information using a variable step size. The core of this method is to adaptively adjust the search step size according to the approximate distance between the current search point and the target surfaces. Using a coarse step size in the area far from the 3D surface and a fine step size in the area close to it can ensure that the 3D surface is fine enough without excess computation.

First, we briefly describe the iterative process for obtaining 3D surface information. The rays from

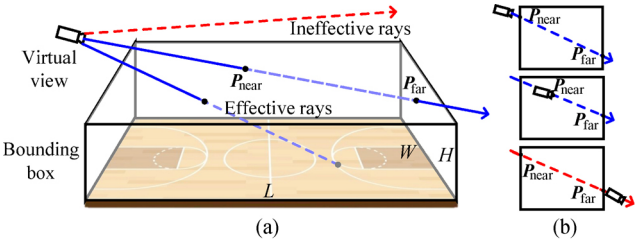

Fig. 2 Search range narrowed by bounding box. (a) Solution for the intersection of the ray and the bounding box. The bounding box encloses a parallelepiped of size $L \times W \times H$. The red line is representative of invalid rays that do not intersect the reconstruction area. Blue lines represent effective rays. (b) Three cases in which the ray intersects the bounding box. Above, middle: there are effective intervals between $\boldsymbol{P}_{\text{near}}$ and $\boldsymbol{P}_{\text{far}}$. Below: the interval is invalid.

the virtual viewpoint are quantized into a number of 3D search points. We project the current search point to all reference images. Then, we count the number of viewpoints which contain projected points in the foreground. If the number exceeds a threshold, the search point belongs to the target surfaces and the search is finished. Otherwise, we use a variable step size to get the next search point and then judge whether it belongs to the target surface. After completing all ray searches we have the 3D surfaces of objects. There are two points to explain. The setting of the threshold should reduce the influence of incomplete segmentation on the rendering result to a certain extent. The adaptive step size modifies the number of 3D sampling points on each ray. Determination of the variable step size is described in detail below. The method mainly includes two steps: finding the closest contour point on a two-dimensional distance field, and a maximum search step in the three-dimensional space based on distance field guidance, as shown in Fig. 3.

Each search point is first projected to every reference viewpoint to obtain its pixel coordinates, which are expressed as $\boldsymbol{p}^n_{\text{cur}} = [u^n_{\text{cur}}, v^n_{\text{cur}}, 1]^{\text{T}}$. If the 3D point belongs to the 3D surface, the search is terminated. Otherwise, we obtain the shortest distance d_n $(n = 1, \ldots, N)$ from the current projection point to the contour of the objects according to the distance field. Then, the search point uses this distance as a step to move along the direction of the projection line. In this way, we can find the projection point of the next search point for the current viewpoint, as shown on the left of Fig. 3. The next search point $\boldsymbol{p}^n_{\text{next}} = [u^n_{\text{next}}, v^n_{\text{next}}, 1]^{\text{T}}$ can be calculated by Eq. (2), where $\boldsymbol{n}$ is a unit vector:

$$\boldsymbol{p}^n_{\text{next}} = \boldsymbol{p}^n_{\text{cur}} + d_n \boldsymbol{n} \tag{2}$$

We obtain the nearest contour points on all reference image planes based on the 2D distance field information. Next, we need to derive the 2D to 3D mapping relationship on the search ray to obtain the variable search step size. The projection relationship between 2D and 3D is key, as shown in Fig. 4.

Each ray emitted from the virtual viewpoint can obtain its starting point $\boldsymbol{P}_{\text{near}}$ and ending point $\boldsymbol{P}_{\text{far}}$ by intersection with the bounding box. Using the internal and external parameters of the camera obtained before, they can be projected onto a two-dimensional plane, giving 2D coordinates $\boldsymbol{p}^n_{\text{near}}$ and $\boldsymbol{p}^n_{\text{far}}$ can be obtained. $\boldsymbol{P}_{\text{near}}$, $\boldsymbol{P}_{\text{far}}$, and $\boldsymbol{P}_{\text{next}}$ are in the same world coordinate system and on the same ray, whose vector parametric equation in space is

$$\boldsymbol{P}_{\text{next}} = t\boldsymbol{P}_{\text{near}} + (1-t)\boldsymbol{P}_{\text{far}} \tag{3}$$

where t is an unknown parameter and $\boldsymbol{P}_{\text{next}}$ is the position of the next search point to be sought. The information in 3D space is insufficient to solve for these two unknowns. Therefore, we transform Eq. (3) to the two-dimensional image plane with more given conditions. We multiply both sides of the equation by the mapping matrix $\boldsymbol{M}_n$; n represents the n-th reference viewpoint. Referring to Eq. (1), it can be concluded that

$$\begin{cases} \boldsymbol{M}_n\boldsymbol{P}_{\text{next}} = t\boldsymbol{M}_n\boldsymbol{P}_{\text{near}} + (1-t)\boldsymbol{M}_n\boldsymbol{P}_{\text{far}} \\ s^n_{\text{next}}\boldsymbol{p}^n_{\text{next}} = ts^n_{\text{near}}\boldsymbol{p}^n_{\text{near}} + (1-t)s^n_{\text{far}}\boldsymbol{p}^n_{\text{far}} \end{cases} \tag{4}$$

where s^n_{near}, s^n_{far}, and s^n_{next} represent the depths of the points in the n-th camera coordinate system. $\boldsymbol{p}^n_{\text{near}} = [u^n_{\text{near}}, v^n_{\text{near}}, 1]^{\text{T}}$, $\boldsymbol{p}^n_{\text{far}} = [u^n_{\text{far}}, v^n_{\text{far}}, 1]^{\text{T}}$, and $\boldsymbol{p}^n_{\text{next}} = [u^n_{\text{next}}, v^n_{\text{next}}, 1]^{\text{T}}$ are the homogeneous coordinates after normalization. Equation (4) may be expanded into a system of linear equations as Eq. (5):

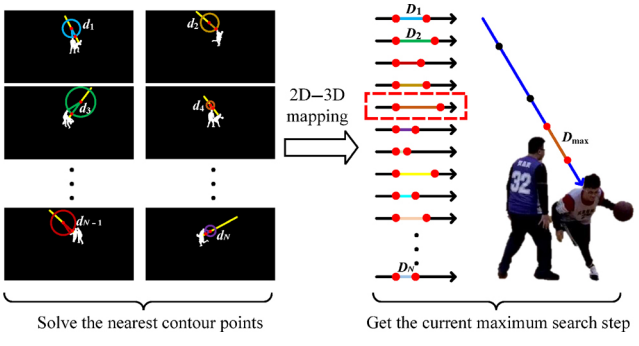

Fig. 3 Variable search step guided by distance field. Left: the different colored line segments indicate the minimum distance between the current point and the contour of the object. Right: after 2D to 3D mapping, the step size on the ray corresponding to different viewpoints is determined, as indicated by the colored line segment. The line segment in the red dashed box is the maximum search step.

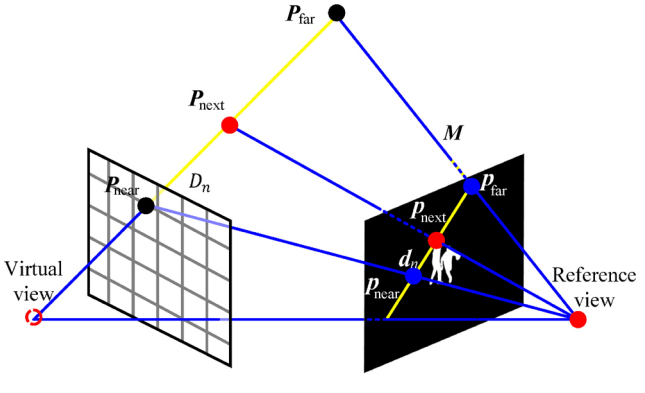

Fig. 4 2D to 3D mapping. The calculation of the next search point on the search ray is based on the guidance of the 2D search step on the distance field.

$$\begin{cases} s^n_{\text{next}} u^n_{\text{next}} = t s^n_{\text{near}} u^n_{\text{near}} + (1-t) s^n_{\text{far}} u^n_{\text{far}} \\ s^n_{\text{next}} v^n_{\text{next}} = t s^n_{\text{near}} v^n_{\text{near}} + (1-t) s^n_{\text{far}} v^n_{\text{far}} \\ s^n_{\text{next}} = t s^n_{\text{near}} + (1-t) s^n_{\text{far}} \end{cases} \tag{5}$$

In the image coordinate system, $\boldsymbol{p}^n_{\text{near}}$, $\boldsymbol{p}^n_{\text{far}}$, and $\boldsymbol{p}^n_{\text{next}}$ are known. The unknowns t and s^n_{next} can be found by simultaneously solving Eq. (5). The expression for unknown parameter t is

$$t = \frac{k_1 - s^n_{\text{far}} u^n_{\text{next}}}{k_2 u^n_{\text{next}} - k_3} \tag{6}$$

where $k_1 = s^n_{\text{far}} u^n_{\text{far}}$, $k_2 = s^n_{\text{near}} - s^n_{\text{far}}$, and $k_3 = s^n_{\text{near}} u^n_{\text{near}} - s^n_{\text{far}} u^n_{\text{far}}$ are parameters to be found. k_1, k_2, and k_3 are only related to the intersections of the rays. These coefficients only need to be calculated once for each ray at the beginning of the iteration. The next search point $\boldsymbol{P}^n_{\text{next}}$ can be calculated by substituting Eq. (6) into Eq. (3). Then, the distance D_n from the current search point is obtained. After all viewpoints have been processed, we obtain a cluster of distance intervals on the ray. Figure 3(right) shows line segments in different colors representing search step sizes fed back by different 2D distance fields. The union of these distance intervals

$$D_{\max} = D_1 \cup D_2 \cup \cdots \cup D_N \tag{7}$$

is the maximum search step, as shown in the red dashed box in Fig. 3. The next search point $\boldsymbol{P}_{\text{next}}$ is acquired by continuing traversal in the ray direction with search step $D_{\max}$.

3.4 Rendering

3.4.1 Concepts

Rendering the 3D surface of the object plays an important role in ensuring the realism of the free-view video. The position and orientation of the desired view may be very different from the reference views, leading to great changes in texture, illumination, color, and occlusion relationships. To achieve realistic and detailed rendering results in the virtual view, reference view ranking and global occlusion judgement based on distance field feedback are carried out for each 3D surface point.

3.4.2 Viewpoint similarity ranking

Diffuse reflection is the most common type of surface reflection in nature. Since the normal directions of the points on a rough surface are inconsistent, reflected light is randomly reflected in different directions. The camera array set up in space samples light from different angles. We believe that the closer the reference viewpoint to the virtual viewpoint, the more realistic the sampled color and texture information will be. We thus use the spatial angle between the virtual and the reference viewpoint as a similarity measure: the smaller the angle, the more similar the viewpoints. In large scenes, the distribution of objects in space is relatively scattered. Therefore, we need to choose the optimal reference view to provide texture for each point on the 3D surfaces to ensure a realistic imaging result.

There is an observation ray between each surface point and the viewpoint. In Fig. 5(left), the line between the surface point and virtual viewpoint is represented by a dashed line. Lines between the surface point and reference viewpoints are shown as solid lines. In Fig. 5, the angle between C_2 and the virtual viewpoint is the smallest and hence most similar.

3.4.3 Global occlusion judgement from distance fields

We choose reference images to provide textures. Besides ranking references by similarity, occlusion of reference views must also be considered. If there is an occluder in the reference view, surface color and texture will be wrong no matter how similar the view is to the virtual view. Sports scenes contain complex occlusion relationships between multiple objects. To determine if a surface point is occluded in the selected view, we need to traverse the ray between that point and the selected view to see if other 3D points have smaller depths. This process is the opposite process to searching for the 3D surface, so the variable-step search guided by the distance field can again be used.

To select the optimal viewpoint for texture information extraction, we sequentially check reference viewpoints for occlusion in order of similarity. We first utilize bounding boxes to narrow the search range. The start point is the surface point on the object, and the end point is determined by the intersection of the distribution area and the ray. Variable-step search using distance field feedback is then carried out. The search point is projected to each reference viewpoint to judge whether it is within the foreground. If the number of times the search point is projected into the foreground is greater than the threshold, then the point is an occluder. C_2 in Fig. 5(right) is very similar to the virtual view, but the color and texture it provides are wrong. We

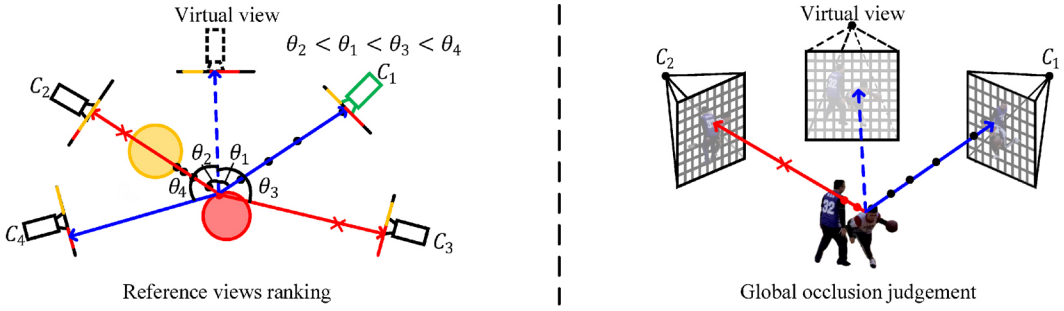

Fig. 5 Left: ranking of reference views. The dotted line camera represents the virtual view. The angle between virtual and reference view rays is used as the basis for ranking basis. Similarity order is given at the upper right. Right: global occlusion judgement from the 3D surface point. Line segments between dots represent variable steps of the distance field feedback. The dotted line represents the virtual view sampling ray. Red rays are invisible. Blue rays indicate visible sampling rays.

do not select this viewpoint and switch to the next viewpoint and continue occlusion checking. If we judge that the current search point does not belong to the occluder, then the distance field information is fused to calculate the next search point. If there is no occlusion in this view, the search along the direction of the reference view will be far away from the object, and the search step will become increasingly larger, showing a divergent trend, as shown by the blue ray in Fig. 5(right). After global occlusion judgement, we select the optimal unoccluded viewing angle to provide texture information for all 3D points on the surface.

3.5 Implementation

We use CUDA to accelerate the algorithm and display the results on the screen using OpenGL. The proposed algorithm has two main parts: distance field-guided variable-step object 3D surface acceleration search and global occlusion judgement. The ray search and judgement of each part are independent. Prepared data are uploaded to the GPU, including multi-view images, silhouettes, distance fields, and camera parameters. Multiple threads are used to compute the 3D surfaces and render the results in parallel. Operations on each ray in our method are independent, so each ray can be computed using a separate thread.

We use the keyboard and mouse to set up the virtual viewpoint. The content of the synthetic virtual view is then drawn on the screen in real time. Users can interactively select any viewpoint to get an immersive 6-DoF visual experience; results of the CUDA parallel processing do not need to be sent back to the CPU but are directly rendered on the screen by the OpenGL.

4 Experiments and analysis

To evaluate the proposed algorithm, we constructed an experimental platform and acquired data from real-world and virtual basketball courts. Using the self-built datasets, allowed us to evaluate visual quality and execution efficiency. In addition, we compared our method to state-of-the-art methods, and analyzed the key factors affecting the algorithm's qualities.

4.1 Experimental setup

4.1.1 Approach

To evaluate our method, we built an experimental platform as shown in Fig. 6. It contained a multi-camera array and multiple computers. The multi-camera array is shown in Fig. 6(bottom, left). We first identified a virtual observation point and then placed all cameras at equal angles centred on that point. Each computer undertakes different tasks, such as acquisition, segmentation, and synthesis. Data are shared between computers through shared files using a local area network. This distributed architecture facilitates task parallelism and scaling, laying the foundation for future free-view video generation systems.

4.1.2 Real-world experimental platform

Care in setting up a suitable experimental platform can not only provide adequate data but also benefit the final imaging results. Camera positioning, image acquisition synchronization, and foreground segmentation all affect the final results for real-world scenes.

Moving away from a controlled laboratory environment, we installed a discrete camera array in an indoor basketball arena to capture real-world data—see Fig. 7(a). Camera installation locations

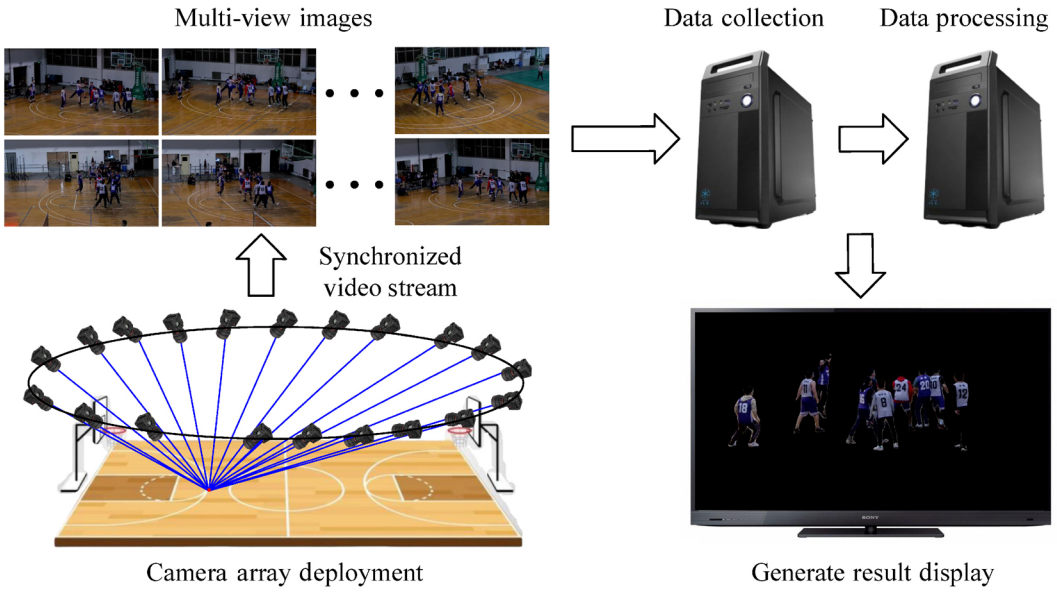

Fig. 6 Free-view video generation system architecture. Lower left: a camera array is set up on a basketball court to acquire synchronous images from multiple perspectives. Lower right: generated free-view video is displayed on the screen.

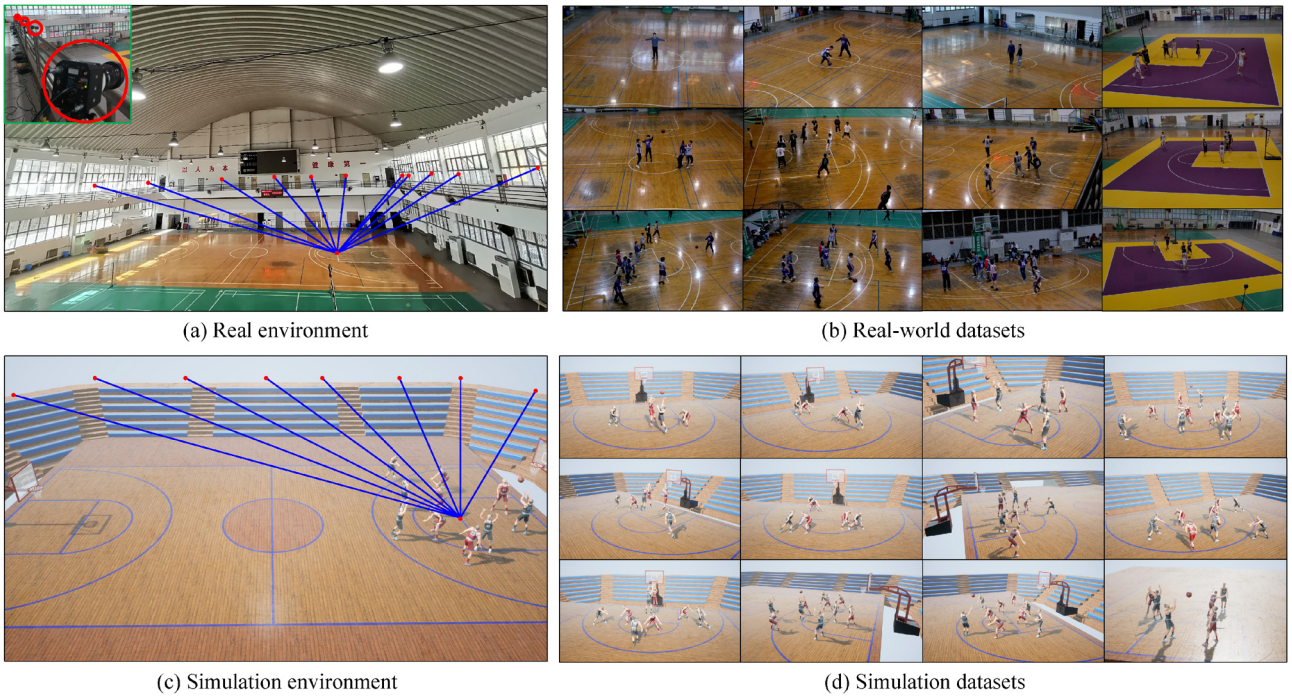

Fig. 7 Construction of real-world and simulated datasets. (a) Real environment. Here and below, the position of the camera is shown by the red dot in the picture. The angle of the camera is shown by the blue line in the figure. (b) Part of the real-world dataset. The figure contains a single person, 1V1, 2V2, 3V3, and 5V5 datasets. (c) Simulated environment. (d) Part of the simulation dataset. The figure contains the 1V1, 2V2, 3V3, 4V4, and 5V5 datasets.

are limited in a sports stadium. Due to space and height constraints, cameras could only be installed on the second floor at roughly evenly distributed angles. The field of view covered an entire half-court area. We used hardware synchronization trigger signals to control acquisition by the camera array, with multiple processes to perform acquisition to ensure the stability of data transmission.

Multi-object segmentation required for data preparation is complex and time-consuming. To obtain better silhouette images, we built a basketball scene dataset to train the segmentation network.

Both accuracy and completeness of the segmentation results are improved using the trained model. Note that the experimental platform is only intended for evaluating the algorithm, and is still some way from being an online free-viewpoint video generation system. Further improvements could, e.g., parallelise the segmentation task, or use more efficient segmentation methods.

4.1.3 Virtual experimental platform

We use the AirSim simulation platform to build a virtual experimental environment, shown in Fig. 7(c). AirSim [51] is a simulator developed by Microsoft in 2017 based on the Unreal Engine to simulate cars and drones. It can acquire sensor poses and transmit scene color and depth data in real time; we innovatively apply it in the evaluation of free-viewpoint synthesis. The simulated environment involves loading the simulated scene, setting internal and external parameters for the camera array, and collecting simultaneous multi-view images.

First, we built a scene model of equal size to the real-world basketball court. It contains a basketball court, basketball hoops, a basketball, and basketball players. Players perform offensive, defensive, shooting, dribbling, and other basketball actions to simulate the complex occlusion relationships in real-world scenes. We then set up 18 viewpoints at equal angles around the simulated scene, with all cameras oriented toward the midpoint of the free-throw line. In addition, we set the fields of view and resolutions of the cameras. Since multiple images cannot be collected at the same time in the simulation, we could only shoot static scenes from multiple viewpoints. This simulation platform not only provides ground truth for internal and external parameters but also offers true color and depth information for other views as a basis for evaluating synthesis quality.

4.1.4 Database

Using the above environments, we constructed real-world and simulation databases to evaluate the proposed algorithm.

We acquired several datasets in the basketball court arena to build a real-world database. It contains multiple groups of multi-view images with different numbers of viewpoints, numbers of targets, and focus areas. The number of viewpoints sampled was 20 and 30. The captured multi-view images contained both single and multiplayer plays, covering a half-court of size 14 m $\times$ 15 m $\times$ 3 m. We classified the data according to the number of targets and viewpoints, such as Single, R-1V1, R-2V2, R-3V3, R-3V3(30), and R-5V5, as shown in Fig. 7(b). Each category contains multiple sets of multi-viewpoint data. The resolution of all multi-viewpoint images is 3840 $\times$ 2160.

We set up 18 sampled viewpoints in the simulated scenes. The field of view again covered a half-court area. We also set up a different number of athletes in the simulations. According to the number of athletes, the simulation database can be divided into the following categories: S-1V1, S-2V2, S-3V3, S-4V4, and S-5V5. Some data from the simulation database are shown in Fig. 7(d). The resolution of the multi-view images is again 3840 $\times$ 2160.

4.1.5 Implementation details

To evaluate our method using large scenes, we compared it to the fixed-step method, instant-NeRF, and Plenoxels. Implementation details for these methods follow.

Instant-NeRF and Plenoxels accelerate the training and rendering of neural radiation fields. We reproduced and modified their source code, running it on an RTX 6000 GPU. Modifications included converting the calibration results in the self-built dataset into the required form, so the parameters used by all methods were the same. Also, we modified them to read multiple view parameters. When training our data with instant-NeRF, we judged whether the current result is optimal by observing the loss graph on the interactive interface. When the parametric model stabilized at the highest value, we stopped training and saved the model. The optimal number of iterations varies with data. It required more than 5000 iterations to train instant-NeRF, taking a few minutes. Plenoxels required 20,000 iterations, taking approximately 30 min.

Our approach and the fixed-step method differ in algorithmic architecture from the NeRF-based approach. We synthesized new viewpoint images without relying on pre-trained models or pre-stored tables, so there is no expensive training time. High-performance GPUs are inessential for the adaptive and fixed-step methods. We evaluated the performance of both methods on a computer with a 3.40 GHz CPU, 32.0 GB RAM, and a GTX 1080Ti GPU. Our method and the fixed-step method

rely on CUDA and OpenGL. In the variable-step search we set the minimum step size to 10 mm, in agreement with the fixed-step method's step size to ensure the same rendering accuracy. Comparing these two methods enables us to evaluate the contribution of variable step size search to algorithm speed. In addition, we executed our method on a computer with an RTX 6000 GPU in order to compare synthesis performance with instant-NeRF and Plenoxels.

4.1.6 Evaluation metrics

We evaluated the effectiveness of our proposed method in terms of execution time and visual quality. For our method and fixed-step synthesis, execution time is determined by two main components: acquiring 3D surfaces and rendering new viewpoint images. We average execution time over multiple frames to eliminate fluctuations caused by different distributions of athletes. The execution time for instant-NeRF and Plenoxels only includes the new view synthesis, not the time to obtain the training model or storage table.

Visual quality is measured by the peak signal to noise ratio (PSNR) and structural similarity (SSIM). Larger PSNR indicates higher imaging performance. SSIM measures image similarity in terms of brightness, contrast, and structure. SSIM values lie in [0, 1]. Larger SSIM indicates lower image distortion.

4.2 Performance evaluation

4.2.1 Setting

In this section, we evaluate the visual quality and synthesis speed of our method on simulated and real-world datasets. Both are affected by the virtual viewpoint location and the number of objects. Therefore, we classify the datasets according to the number of objects and quantify the distance between the virtual viewpoint and the targets. In the half-court, the distance is quantified as the distance from the novel viewpoint to the centre point of the free-throw line. In the mid-court, the distance is quantified as the distance from the novel viewpoint to the centre point of the mid-court line. Medium distance here corresponds to 12–15 m.

4.2.2 Visual quality evaluation

We categorize the database according to the number of sampled viewpoints and objects. In the real-world database, there are R-Single, R-1V1, R-2V2, R-3V3, R-3V3(30), and R-5V5. The datasets for the simulation basketball scene are S-1V1, S-2V2, S-3V3, S-4V4, and S-5V5. Each dataset contains multiple sets of multi-view images with the same number of objects and different distributions. We now evaluate the visual quality of the proposed method using these datasets.

Figures 8 and 9 show novel viewpoint generation results for dynamic objects in real-world and simulated scenes, respectively. Close-ups are provided in red boxes, showing details such as patterns, text, and folds, on the clothes; we can also see human details such as the face, muscle lines, and fingers. These enlarged details show that the surface texture of the rendered image is smooth and continuous.

Comparing results from the simulated and the real-world datasets, we see that synthesised edges of objects in the real-world datasets are not as smooth as

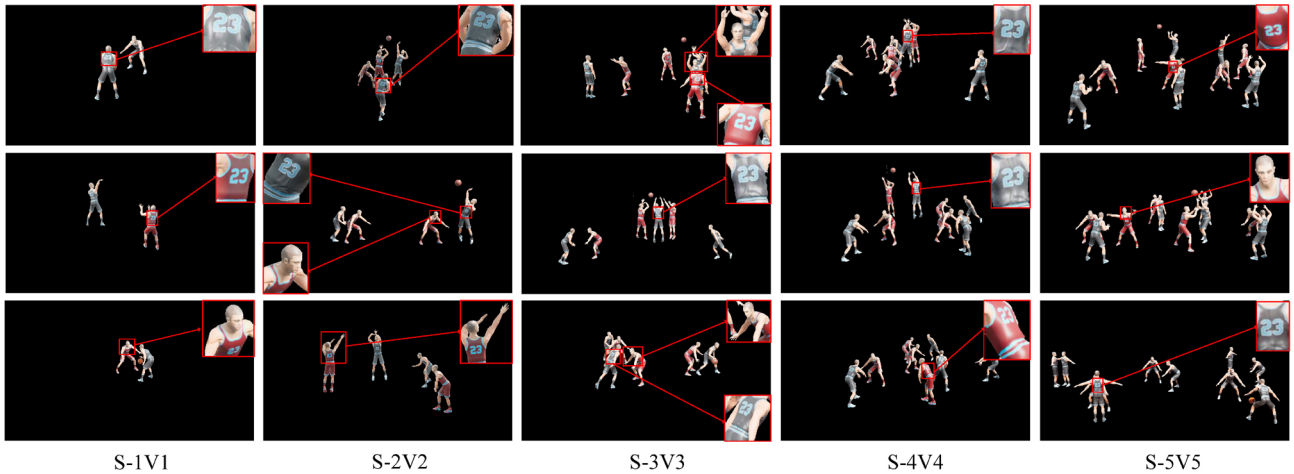

Fig. 8 Imaging results using simulation data with the virtual viewpoint at a medium distance. Each column contains a different number of objects. Details of clothing texture, text, faces, etc., are enlarged in red boxes.

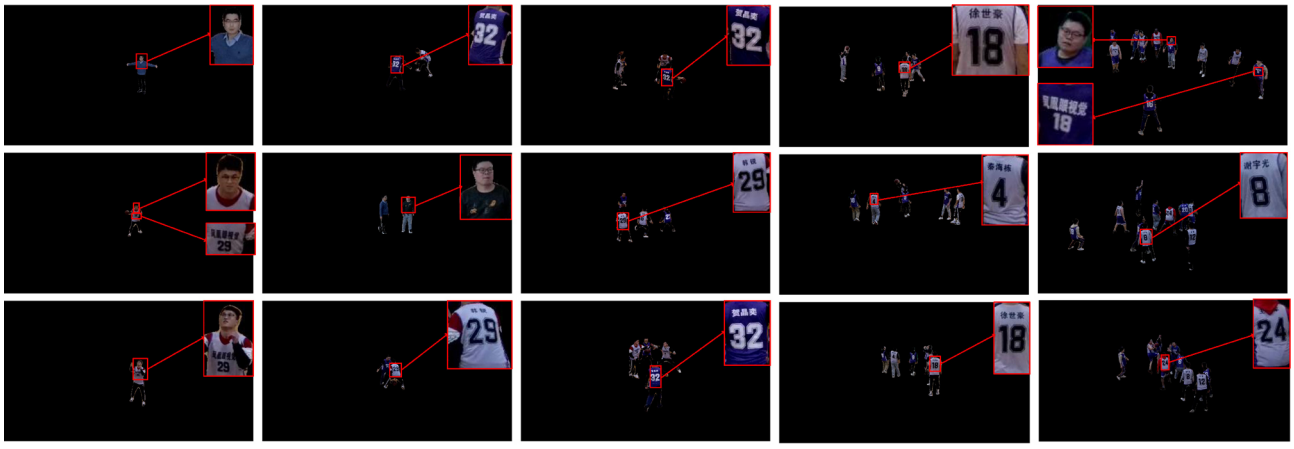

Fig. 9 Imaging results on the real-world dataset with the virtual viewpoint at a medium distance. Each column contains a different number of objects. Details are again displayed in the red boxes.

for the simulation datasets. The simulation platform provides ground truth multi-view segmentations and camera parameters. However, segmentations of objects in the real-world environment are obtained by deep learning, with inaccurate fitting of object contours and inclusion of part of the background. In addition, occlusion between objects is more complicated in the real-world environment. There is not only mutual occlusion between objects but also occlusion by static objects and self-occlusion. Nevertheless, the method proposed in this paper still produces good results.

4.2.3 Execution time evaluation

The execution time of our method and the fixed-step method on real-world and simulation datasets is reported in Tables 1 and 2, respectively, using a GTX 1080 Ti GPU. The resolution of the synthesised images was 1920 × 1080. Time is given for 3D surface estimation (1st step), rendering (2nd step), and in total. Other data transfer and preprocessing time is not included. To eliminate the influence of viewpoint location and object distribution, we set virtual viewpoints at different locations within a medium distance and report averages.

Table 1 Execution time per frame (ms) for real-world data

Dataset	Ours			Fixed-step			Speed-up
	1st	2nd	Total	1st	2nd	Total	
R-Single	14.60	2.90	17.84	245.00	6.01	251.45	14.09×
R-1V1	15.11	4.29	19.74	265.73	12.86	279.04	14.14×
R-2V2	20.94	7.95	29.23	298.75	21.50	320.65	10.97×
R-3V3	18.81	14.13	33.27	391.35	58.45	450.20	13.53×
R-5V5	36.78	28.63	65.74	399.89	139.51	539.81	8.21×

Table 2 Execution time per frame (ms) for simulated data

Dataset	Ours			Fixed-step			Speed-up
	1st	2nd	Total	1st	2nd	Total	
S-1V1	15.93	4.81	21.07	305.16	21.13	326.69	15.51×
S-2V2	22.22	9.03	31.59	331.54	31.47	363.46	11.51×
S-3V3	27.85	11.76	40.00	428.83	72.38	501.72	12.54×
S-4V4	33.88	16.76	50.99	460.15	104.78	565.53	11.09×
S-5V5	35.11	20.95	56.41	522.67	157.12	680.32	12.06×

As the tables show, the rendering speed of our method is significantly improved by using variable step size compared to the fixed-step method. Speeds of both acquisition of 3D surfaces and image rendering are improved. It can also be observed that the speed of new viewpoint image synthesis is affected by the number of objects. When there are few objects, our method can reach about 50 fps; when the number is larger, the speed is close to 20 fps. For fewer than 6 objects, realizing real-time synthesis can be achieved in a volume of 14 m × 15 m × 3 m.

4.3 Comparative tests

To evaluate our method more broadly, we compare it to the fixed-step method and state-of-the-art view synthesis methods based on neural rendering, including instant-NeRF and Plenoxels. The fixed-step method is similar to our method except that its ray search uses a fixed step size, so a comparison to it can verify the utility of our proposed variable-step strategy. Instant-NeRF greatly improves the training speed of the network, reducing the training time to minutes or even seconds. Its rapid training and excellent display results have attracted much

attention. Plenoxels represents the scene as a sparse 3D grid with spherical harmonics. This method uses only a differentiable volume renderer to obtain realistic compositions without any neural component. Our comparisons consider visual quality and execution speed.

To compare synthetic image quality, we selected 12 representative sets from the self-built database. We randomly selected one viewpoint as a test in each set, and the remaining images provided the information to synthesize test viewpoints. For a fair comparison, instant-NeRF and Plenoxels shared the same multi-view dataset and calibration parameters as our approach for training. Qualitative experimental results are shown in Fig. 10. Table 3 summarizes the PSNR and SSIM relating the rendered results to the ground truth. All results were rendered at 3840 pixels × 2160 pixels for comparison to the reference image.

Fig. 10 Rendering results from various methods based on real-world datasets. Close-ups show that our approach provides more detail than instant-NeRF and Plenoxels. Green boxes give ground truth, and red boxes are rendered results. In many cases Plenoxels failed to render a view, which we indicate with a red cross. Floaters in rendered results are indicated by blue dashed boxes.

Table 3 Rendering quality compared to ground truth for our method, the fixed-step method, instant-NeRF, and Plenoxels, using the self-built database

Dataset	Ours		Fixed-step		Instant-NeRF		Plenoxels	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
R-1V1	**40.068**	**0.996**	40.029	**0.996**	35.522	0.983	15.346	0.963
R-2V2	**34.773**	**0.990**	34.760	**0.990**	28.180	0.956	17.448	0.963
R-3V3	**32.475**	**0.982**	32.470	**0.982**	27.139	0.905	13.298	0.926
R-3V3(30)	37.857	0.991	**38.314**	**0.992**	31.904	0.973	27.728	0.981
R-5V5	30.808	**0.968**	**31.019**	**0.968**	25.027	0.870	10.215	0.847

Figure 10 shows the rendered results and ground-truth. Our method always faithfully restores the positions and textures of objects. While the results of instant-NeRF contain correct texture and position information, they also include some floating objects. Plenoxels requires a relatively high number of sampling viewpoints. When the number of sampled viewpoints is small, Plenoxels cannot reproduce the original picture and produces some ghosting. These results demonstrate that our method is superior to instant-NeRF and Plenoxels in terms of rendered image realism, regardless of the number of sampled views or objects. Text, faces, and other details rendered by our method are closer to the ground truth. For NeRF-based methods, the datasets used do not have enough sampled viewpoints to predict the exact 3D positions of the objects, so ghosts or floaters are generated. However, our method incorporates multi-view geometric information to effectively filter out non-target points and suppress floating objects. Furthermore, we choose the reference image most similar to the virtual viewpoint to provide texture information; this reference view must also pass a global occlusion test. Accurate estimation of the target 3D surface and texture rendering using the optimal viewing angle make our renderings more realistic.

The PSNR and SSIM linking the rendering results to the ground truth are shown in Table 3. The rendering quality of our method is comparable to that of the fixed-step method, showing that our proposed adaptive step does not reduce picture quality. Furthermore, the rendering results of our method are significantly better than those of instant-NeRF and Plenoxels. R-3V3 and R-3V3(30) contain the same number of targets, but the latter has 10 more sampling viewpoints. We can see a significant numerical improvement for all methods when the number of viewpoints increases. Furthermore, we can see that the PSNR and SSIM of all methods decreases gradually as the number of objects increases (except for R-3V3(30)). On the one hand, as the number of targets increases, the occlusion relationships between objects become more complex. Sparse viewpoints are insufficient to provide information to recover accurate 3D positions. On the other hand, having more targets means that the proportion of foreground pixels increases, which also causes numerical changes. Multiple targets and sparsely viewpoints are not conducive to new viewpoint image synthesis. Even with a large number of objects and sparse collection viewpoints, the PSNR of our method is still greater than 30.

We performed two further comparisons to evaluate the execution speed of the proposed method, a comparison to the fixed-step method based on real-world and simulated datasets, and a further comparison to the fixed-step, instant-NeRF, and Plenoxels methods. We set the rendering resolution to 1920×1080 and the virtual viewpoint position within the medium distance range.

New viewpoint synthesis time for our method and the fixed-step method is given in Tables 1 and 2. They show that 3D surface generation is the most time-consuming part of our method and the fixed-step method, accounting for about 68% and 87% of the total execution time, respectively. The more time-consuming nature of acquiring the 3D surface is due to the fact that the number of search rays is consistent with the rendering resolution and the search range is larger. However, ray search in the rendering stage only starts from the target surface, so the number of rays and search range are smaller. The smaller proportion of acquiring 3D surfaces in our method reflects that variable step sizes guided by distance fields can effectively speed up the ray search.

We see that using a variable step size greatly improves the execution speed compared to the fixed-step method. While execution takes longer for more object, our method is 8–15 times faster than the fixed-step method.

In addition, we evaluated the new viewpoint synthesis speed for our method, the fixed-step method, instant-NeRF, and Plenoxels using real-world data, with results shown in Table 4. NeRF-based methods typically require high-performance GPUs, so we

Table 4 Execution time per frame (ms) for our method, the fixed step method, instant-NeRF, and Plenoxels on the self-built database. Time for instant-NeRF and Plenoxels does not include the time to for training or constructing a storage table

Dataset	Ours (GTX 1080 Ti)	Fixed-step (GTX 1080 Ti)	Ours (RTX 6000)	Instant-NeRF (RTX 6000)	Plenoxels (RTX 6000)
R-1V1	19.74	279.04	**17.95**	27.65	208.43
R-2V2	29.23	320.65	**23.86**	33.45	230.30
R-3V3	33.27	450.20	**26.62**	31.99	271.74
R-3V3(30)	46.59	486.36	34.97	**34.41**	269.04
R-5V5	65.74	539.81	**40.34**	40.86	264.49

trained instant-NeRF and Plenoxels using a Quadro RTX 6000 GPU. However, our method does not require such large memory resources. For a fuller evaluation, we measured the synthesis time of our method on both a GTX 1080 Ti GPU and a Quadro RTX 6000 GPU. Execution time for our method and the fixed-step method includes 3D surface estimation and image rendering. Execution time for instant-NeRF and Plenoxels only includes new view synthesis, not the time for training or constructing the storage table.

The first two columns of Table 4 show that our method is (8–15)× faster than the fixed-step method on the GTX 1080 Ti GPU, demonstrating the effectiveness of the variable-step search strategy. We compare execution time of our method to those of instant-NeRF and Plenoxels on the same hardware configuration in the last three columns of Table 4. Overall, our method is the fastest, instant-NeRF is slower, and Plenoxels is the slowest. Furthermore, our execution speed on a GTX 1080 Ti GPU is higher than that of instant-NeRF and Plenoxels in the case of few targets, as shown by rows R-1V1 and R-2V2 in Table 4. Our method outperforms the fixed-step method, instant-NeRF, and Plenoxels, given the same hardware configuration. Moreover, our method is efficient even on a consumer-grade graphics card like the GTX 1080 Ti GPU.

4.4 Effects and analyses

We now further explore the effect of the number of targets and virtual viewpoints on the speed of variable-step search. Our goal is to improve synthesis speed without losing image quality, so we also compare visual quality to that of the fixed-step approach while making these changes.

4.4.1 Effect on speed

Execution speed is related to the number of effective rays (rays containing targets), which in turn is affected by the number of objects and the position of the virtual viewpoint. To explore the impact of the above factors on execution time, we provide histograms for different numbers of targets and viewpoint positions in Figs. 11 and 12.

We set the viewpoint at a medium distance and measured the execution time for different numbers of objects. Viewpoint positions were not the same for different datasets. To eliminate the influence of virtual viewpoint position, we set the viewpoint to a series of positions and averaged the execution time. Figure 11 shows execution time for the real-world and simulation databases. Each group of experimental data corresponds to two sets of execution time. The left bar represents our approach and the right bar represents the fixed-step method. Different colors represent different execution steps. The execution time becomes longer as the number of objects increases for both methods, for both 3D surface estimation and rendering.

There are two main reasons why the number of objects affects the execution time. On the one hand, the number of effective rays increases with the number of objects when virtual viewpoints are set at the same distance. We project the 3D point on the search ray onto the silhouette maps to determine whether it belongs to the target surfaces. More search rays mean that more judgements need to be made. The number of global occlusion judgements in the rendering stage also increases. While our method reduces the number of search steps in each ray, the total number of rays is determined by the content. On the other hand, as the number of objects increases, the occlusion relationship between objects becomes more complicated. The complex occlusion relationships complicate the distance field as well. The adaptive step size will also be reduced accordingly. Interleaving of multiple objects also affects the global occlusion judgement. It takes longer to choose the optimal viewing angle during the rendering phase. Therefore, the rendering efficiency of the proposed method decreases as the number of objects increases. Nevertheless, regardless of the number of targets, our approach always achieves an (8–15)× improvement in speed.

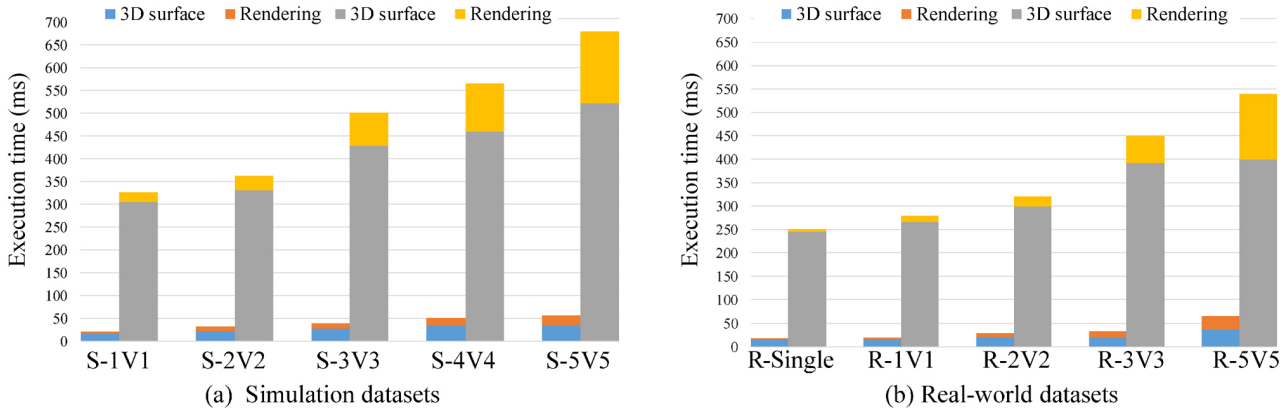

Fig. 11 Execution time for our method and the fixed-step method, for different datasets.

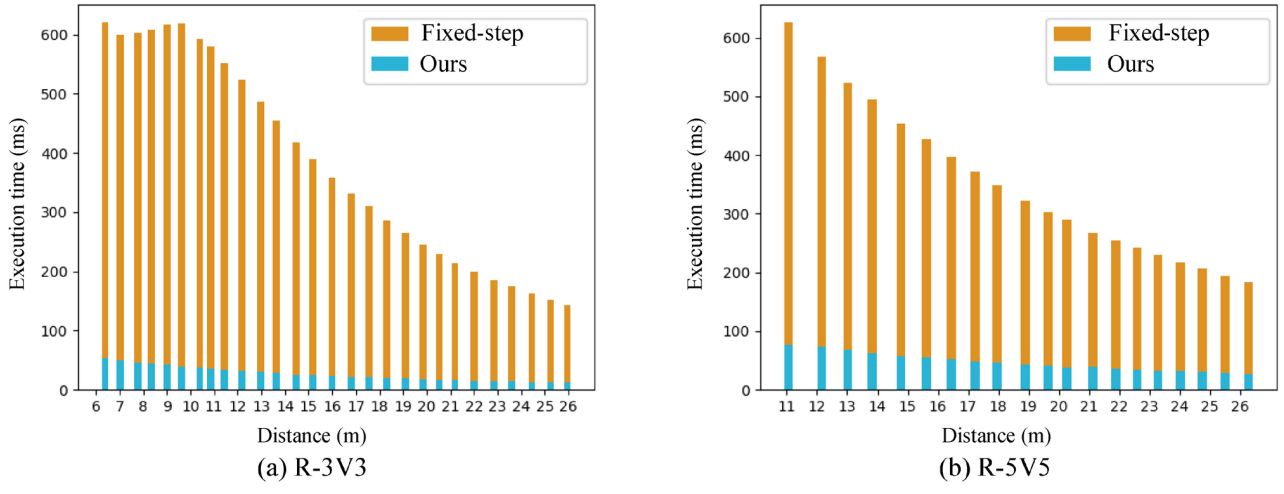

Fig. 12 Execution time for our method and the fixed-step method, at different distances.

To explore the relationship between the virtual viewpoint position and execution time, we selected two groups from the real-world datasets. A fixed virtual camera angle was set for each group, and only the position changed to record the execution time. The distance between the virtual viewpoint and the objects was between 5 and 30 m. We again plot the results as histograms, shown in Fig. 12. The orange bars record synthesis time for the fixed-step method at different positions. The blue bars record time for our method. Execution time of both methods decrease as the virtual viewpoint recedes from the targets: fewer effective rays are involved in rendering, reducing the calculation needed for 3D surface estimation and image rendering. Thus synthesis speed increases with distance. In Fig. 12(a), when the virtual viewing distance is small, execution time for the fixed-step method fluctuates, perhaps because the say search range becomes shorter, and invalid rays exit the search quickly.

These results indicate that the closer the virtual viewpoint is to the objects, the slower the rendering speed. Nevertheless, regardless of distance, our rendering speed is significantly improved compared to the fixed-step method. Our method is particularly applicable to large scenes with long viewpoint distances and relatively small targets.

4.4.2 Effect on visual quality

We next explore the influence of the number of objects and the position of the virtual viewpoint on visual quality.

We first chose two datasets containing different object numbers from real-world datasets. Then, we set virtual viewpoints at different distances to evaluate the effect of position on visual quality. Two groups of imaging results are presented in Fig. 13, R-3V3 and R-5V5. The upper row of each group shows 3D surface estimation and rendering results of our method; the bottom row shows the results of the fixed-step method. The first two columns show

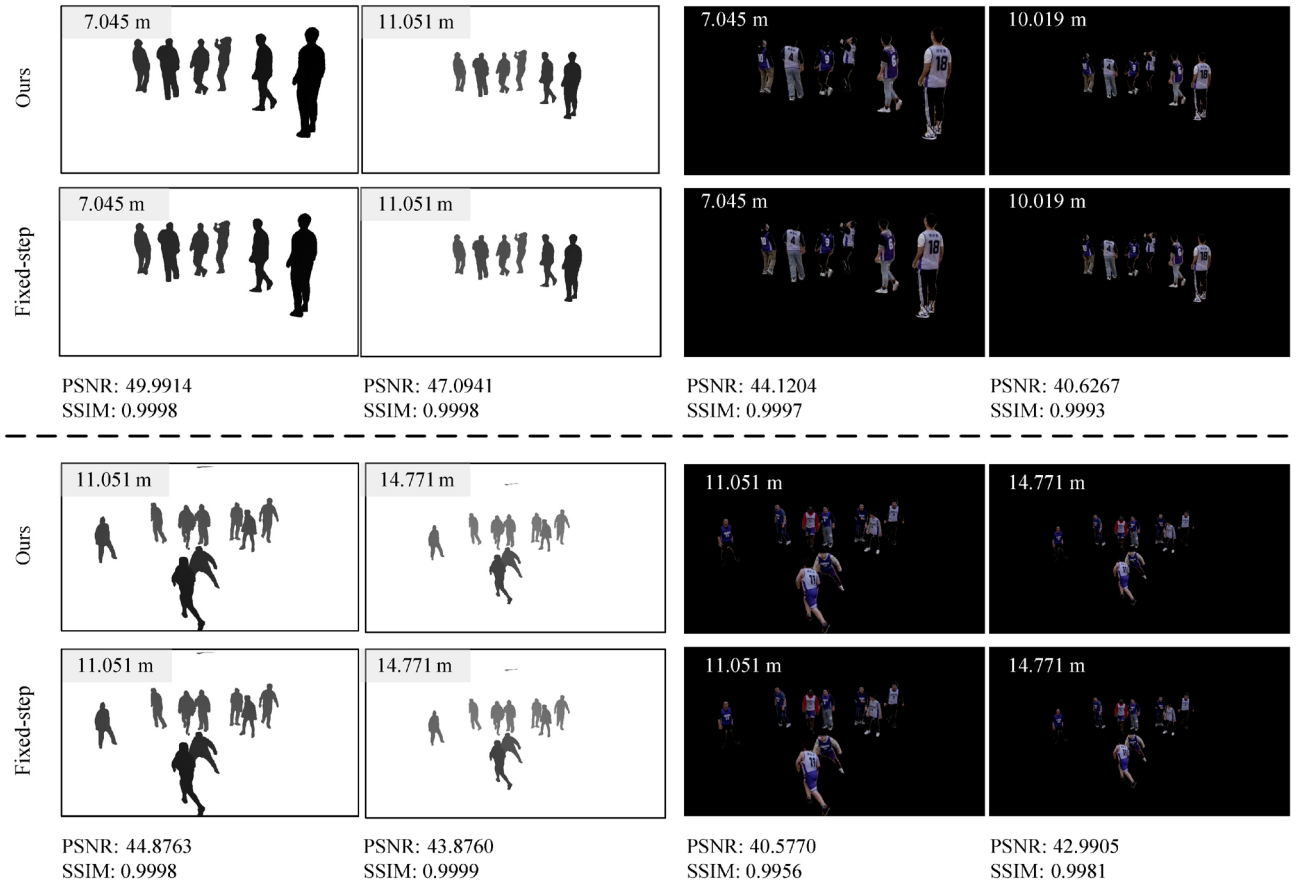

Fig. 13 Comparison of our method to the fixed-step method in 3D surface estimation and rendering. Above: using the R-3V3 dataset. Below: using R-5V5.

the multi-objective 3D surface. We normalize the surface depth (in the range of 5–30 m) and transform it to grayscale for display. The last two columns show the final rendering results. The distance between the virtual viewpoint and the objects is given in the upper left corner of each picture. The PSNR value is above 40, and the SSIM value is close to 0.99; the imaging quality of our method is the same as that of the fixed-step method.

Our proposed new viewpoint synthesis method based on distance field acceleration fuses distance field information from all viewpoints and utilizes it to guide variable-step searches. When the distance from the target is far, we use a large step size to quickly cross the no-target area. When the distance to the target is very small, the search step size becomes very fine. To avoid the inefficiency of it being too fine, we set the minimum step size to 10 mm, which is consistent with the fixed step size. The finest step size determines the quality of the rendered result. Therefore, the image rendering quality of our method is always consistent with the fixed-step method, while reducing the amount of searching, as these results confirm.

5 Limitations and future work

Although our method can satisfy multi-objective real-time rendering of large-scale scenes, the quality of rendering is still related to the accuracy of segmentation. We explored the effect of different segmentation failures on rendering results based on the simulation datasets, as shown in Fig. 14. While our method can remove some spurious regions, it cannot completely overcome silhouette expansion or missing content. We set a threshold for judging 3D target surfaces to reduce the impact of incomplete segmentation results. However, using the same threshold is not suitable for all scenarios. In future, we hope to consider further improving the accuracy of segmentation, and reducing the dependence of the synthesis method on the segmentation result.

A real-time online interactive free-view video

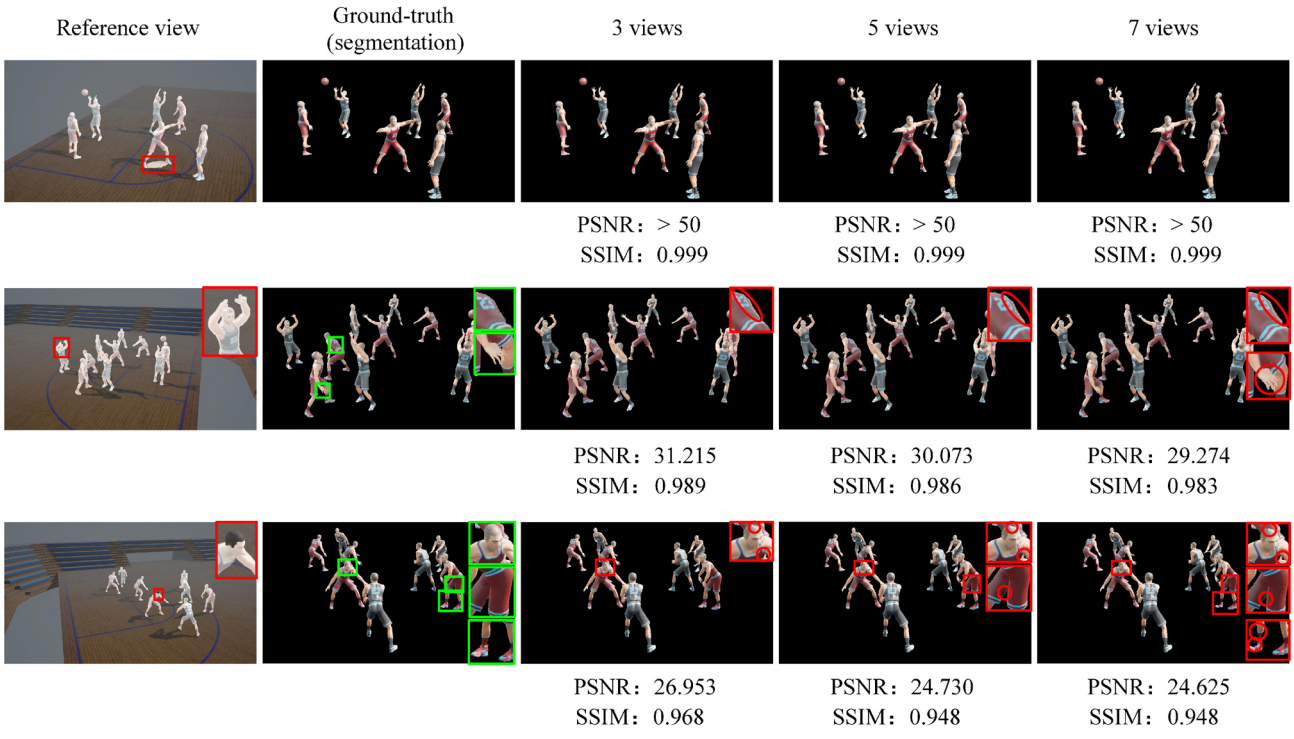

Fig. 14 Effect of segmentation accuracy on our method, using simulated data. Incorrect segmentation takes three forms: the foreground contains wrong object pixels (top), the foreground region extends over an object edge (middle), or the foreground region lacks object pixels (bottom).

generation system can provide an immersive viewing experience. Free-view video synthesis is the key to real-time interactive systems. Our proposed method can meet the requirements of synthesizing new viewpoints in real time. However, there are still many other steps needed to provide online applications, including multi-viewpoint video collection, data transmission, and data processing. We have built an experimental platform, but it is still some distance from online applications. Other possible research directions are to explore free-viewpoint data compression, transfer, and processing.

6 Conclusions

In large scenes, factors such as broad coverage, multiple objects, and sparse sampling viewpoints reduce the speed and sharpness of new viewpoint synthesis. To address these issues, we have proposed a variable-step search based on distance field guidance to speed up synthesis without loss of visual quality in synthesized images. We fuse multi-view distance fields and use them to guide the adaptive step search. The search rays emitted from the virtual viewpoint use a variable step to quickly cross empty volumes, reducing the time to acquire multi-object 3D surfaces. Global occlusion judgement guided by the distance field helps to rapidly select the optimal view to provide texture information. We have implemented our ideas in CUDA and OpenGL, allowing an extensive evaluation of our approach on both simulated and real-world data, and comparisons to state-of-the-art methods. The experimental results demonstrate that our method provides an improvement in speed by at least 8 times over the fixed-step method, without loss of visual quality. For large sports scenes, our method can synthesize new viewpoint images with a resolution of 1920 × 1080 at 25 fps.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 62172315, 62073262, and 61672429), the Fundamental Research Funds for the Central Universities, the Innovation Fund of Xidian University (No. 20109205456), the Key Research and Development Program of Shaanxi (No. S2021-YF-ZDCXL-ZDLGY-0127), and HUAWEI.

Author contributions

Yanran Dai made a significant contribution to the design and implementation of the methodology, the analysis of the data, and the writing of the manuscript. Jing Li contributed significantly to the conception of the study and reviewed and revised the first draft. Yuqi Jiang, Haidong Qin, and Haozhe Pan set up the real-world and simulation experimental environment and collated the experimental data. Bang Liang and Shikuan Hong designed and validated the experiments. Tao Yang helped with the analysis and gave constructive comments.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Electronic Supplementary Material

Supplementary material is available in the online version of this article at https://doi.org/10.1007/s41095-022-0323-3.

References

[1] Fukushima, N.; Fujii, T.; Ishibashi, Y.; Yendo, T.; Tanimoto, M. Real-time free viewpoint image rendering by using fast multi-pass dynamic programming. In: Proceedings of the 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, 1–4, 2010.

[2] Wang, R. G.; Luo, J. J.; Jiang, X. B.; Wang, Z. Y.; Wang, W. M.; Li, G.; Gao, W. Accelerating image-domain-warping virtual view synthesis on GPGPU. *IEEE Transactions on Multimedia* Vol. 19, No. 6, 1392–1400, 2017.

[3] Ceulemans, B.; Lu, S. P.; Lafruit, G.; Munteanu, A. Robust multiview synthesis for wide-baseline camera arrays. *IEEE Transactions on Multimedia* Vol. 20, No. 9, 2235–2248, 2018.

[4] Cheung, C. H.; Sheng, L.; Ngan, K. N. Motion compensated virtual view synthesis using novel particle cell. *IEEE Transactions on Multimedia* Vol. 23, 1908–1923, 2021.

[5] Nonaka, K.; Sabirin, H.; Chen, J.; Sankoh, H.; Naito, S. Optimal billboard deformation via 3D voxel for free-viewpoint system. *IEICE Transactions on Information and Systems* Vol. E101.D, No. 9, 2381–2391, 2018.

[6] Sabirin, H.; Yao, Q.; Nonaka, K.; Sankoh, H.; Naito, S. Toward real-time delivery of immersive sports content. *IEEE MultiMedia* Vol. 25, No. 2, 61–70, 2018.

[7] Sankoh, H.; Naito, S.; Nonaka, K.; Sabirin, H.; Chen, J. Robust billboard-based, free-viewpoint video synthesis algorithm to overcome occlusions under challenging outdoor sport scenes. In: Proceedings of the 26th ACM International Conference on Multimedia, 1724–1732, 2018.

[8] Yao, Q.; Nonaka, K.; Sankoh, H.; Naito, S. Robust moving camera calibration for synthesizing free viewpoint soccer video. In: Proceedings of the IEEE International Conference on Image Processing, 1185–1189, 2016.

[9] Chen, J.; Watanabe, R.; Nonaka, K.; Konno, T.; Sankoh, H.; Naito, S. A robust billboard-based free-viewpoint video synthesizing algorithm for sports scenes. *arXiv preprint* arXiv:1908.02446, 2019.

[10] Yamada, K.; Sankoh, H.; Sugano, M.; Naito, S. Occlusion robust free-viewpoint video synthesis based on inter-camera/-frame interpolation. In: Proceedings of the IEEE International Conference on Image Processing, 2072–2076, 2013.

[11] Shin, T.; Kasuya, N.; Kitahara, I.; Kameda, Y.; Ohta, Y. A comparison between two 3D free-viewpoint generation methods: Player-billboard and 3D reconstruction. In: Proceedings of the 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, 1–4, 2010.

[12] Carballeira, P.; Carmona, C.; Díaz, C.; Berjón, D.; Corregidor, D.; Cabrera, J.; Morán, F.; Doblado, C.; Arnaldo, S.; del Mar Martín, M.; et al. FVV live: A real-time free-viewpoint video system with consumer electronics hardware. *IEEE Transactions on Multimedia* Vol. 24, 2378–2391, 2022.

[13] Hedman, P.; Philip, J.; Price, T.; Frahm, J. M.; Drettakis, G.; Brostow, G. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 257, 2018.

[14] Do, L.; Bravo, G.; Zinger, S.; De With, P. H. N. GPU-accelerated real-time free-viewpoint DIBR for 3DTV. *IEEE Transactions on Consumer Electronics* Vol. 58, No. 2, 633–640, 2012.

[15] Gao, X. S.; Li, K. Q.; Chen, W. Q.; Yang, Z. Y.; Wei, W. G.; Cai, Y. G. Free viewpoint video synthesis based on DIBR. In: Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval, 275–278, 2020.

[16] Li, S.; Zhu, C.; Sun, M. T. Hole filling with multiple reference views in DIBR view synthesis. *IEEE Transactions on Multimedia* Vol. 20, No. 8, 1948–1959, 2018.

[17] Tian, S. S.; Zhang, L.; Morin, L.; Déforges, O. A benchmark of DIBR synthesized view quality assessment metrics on a new database for immersive media

applications. *IEEE Transactions on Multimedia* Vol. 21, No. 5, 1235–1247, 2019.

[18] Wang, X. J.; Shao, F.; Jiang, Q. P.; Meng, X. C.; Ho, Y. S. Measuring coarse-to-fine texture and geometric distortions for quality assessment of DIBR-synthesized images. *IEEE Transactions on Multimedia* Vol. 23, 1173–1186, 2021.

[19] Jin, J.; Wang, A. H.; Zhao, Y.; Lin, C. Y.; Zeng, B. Region-aware 3-D warping for DIBR. *IEEE Transactions on Multimedia* Vol. 18, No. 6, 953–966, 2016.

[20] Liu, Z. M.; Jia, W.; Yang, M.; Luo, P. Y.; Guo, Y.; Tan, M. K. Deep view synthesis via self-consistent generative network. *IEEE Transactions on Multimedia* Vol. 24, 451–465, 2022.

[21] Zhou, T. H.; Tucker, R.; Flynn, J.; Fyffe, G.; Snavely, N. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics* Vol. 37, No. 4, Article No. 65, 2018.

[22] Wang, Y. R.; Huang, Z. H.; Zhu, H.; Li, W.; Cao, X.; Yang, R. G. Interactive free-viewpoint video generation. *Virtual Reality & Intelligent Hardware* Vol. 2, No. 3, 247–260, 2020.

[23] Broxton, M.; Flynn, J.; Overbeck, R.; Erickson, D.; Hedman, P.; Duvall, M.; Dourgarian, J.; Busch, J.; Whalen, M.; Debevec, P. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics* Vol. 39, No. 4, Article No. 86, 2020.

[24] Broxton, M.; Busch, J.; Dourgarian, J.; DuVall, M.; Erickson, D.; Evangelakos, D.; Flynn, J.; Hedman, P.; Overbeck, R.; Whalen, M.; et al. DeepView immersive light field video. In: Proceedings of the ACM SIGGRAPH Immersive Pavilion, Article No. 15, 2020.

[25] Flynn, J.; Broxton, M.; Debevec, P.; DuVall, M.; Fyffe, G.; Overbeck, R.; Snavely, N.; Tucker, R. DeepView: View synthesis with learned gradient descent. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2362–2371, 2019.

[26] Penner, E.; Zhang, L. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics* Vol. 36, No. 6, Article No. 235, 2017.

[27] Dou, M. S.; Khamis, S.; Degtyarev, Y.; Davidson, P.; Fanello, S. R.; Kowdle, A.; Escolano, S. O.; Rhemann, C.; Kim, D.; Taylor, J.; et al. Fusion4D: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics* Vol. 35, No. 4, Article No. 114, 2016.

[28] Wei, D. X.; Xu, X. W.; Shen, H. B.; Huang, K. J. GAC-GAN: A general method for appearance-controllable human video motion transfer. *IEEE Transactions on Multimedia* Vol. 23, 2457–2470, 2021.

[29] Collet, A.; Chuang, M.; Sweeney, P.; Gillett, D.; Evseev, D.; Calabrese, D.; Hoppe, H.; Kirk, A.; Sullivan, S. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 69, 2015.

[30] Huang, Z.; Li, T. Y.; Chen, W. K.; Zhao, Y. J.; Xing, J.; LeGendre, C.; Luo, L. J.; Ma, C. Y.; Li, H. Deep volumetric video from very sparse multi-view performance capture. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11220*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 351–369, 2018.

[31] Natsume, R.; Saito, S.; Huang, Z.; Chen, W. K.; Ma, C. Y.; Li, H.; Morishima, S. SiCloPe: Silhouette-based clothed people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4475–4485, 2019.

[32] Leroy, V.; Franco, J. S.; Boyer, E. Volume sweeping: Learning photoconsistency for multi-view shape reconstruction. *International Journal of Computer Vision* Vol. 129, No. 2, 284–299, 2021.

[33] Zheng, Z. R.; Yu, T.; Liu, Y. B.; Dai, Q. H. PaMIR: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 44, No. 6, 3170–3184, 2022.

[34] Meerits, S.; Thomas, D.; Nozick, V.; Saito, H. FusionMLS: Highly dynamic 3D reconstruction with consumer-grade RGB-D cameras. *Computational Visual Media* Vol. 4, No. 4, 287–303, 2018.

[35] Li, J. W.; Gao, W.; Wu, Y. H.; Liu, Y. D.; Shen, Y. F. High-quality indoor scene 3D reconstruction with RGB-D cameras: A brief review. *Computational Visual Media* Vol. 8, No. 3, 369–393, 2022.

[36] Nonaka, K.; Watanabe, R.; Chen, J.; Sabirin, H.; Naito, S. Fast plane-based free-viewpoint synthesis for real-time live streaming. In: Proceedings of the IEEE Visual Communications and Image Processing, 1–4, 2018.

[37] Yusuke, U.; Takahashi, K.; Fujii, T. Free viewpoint video generation system using visual hull. In: Proceedings of the International Workshop on Advanced Image Technology, 1–4, 2018.

[38] Chen, J.; Watanabe, R.; Nonaka, K.; Konno, T.; Sankoh, H.; Naito, S. Fast free-viewpoint video synthesis algorithm for sports scenes. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 3209–3215, 2019.

[39] Watanabe, T.; Tanaka, T. Free viewpoint video synthesis on human action using shape from silhouette method. In: Proceedings of the SICE Annual Conference, 2748–2751, 2010.

[40] Dellaert, F.; Yen-Chen, L. Neural volume rendering: NeRF and beyond. *arXiv preprint* arXiv:2101.05204, 2020.

[41] Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 405–421, 2020.

[42] Lombardi, S.; Simon, T.; Schwartz, G.; Zollhoefer, M.; Sheikh, Y.; Saragih, J. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics* Vol. 40, No. 4, Article No. 59, 2021.

[43] Yu, A.; Li, R. L.; Tancik, M.; Li, H.; Ng, R.; Kanazawa, A. PlenOctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5732–5741, 2021.

[44] Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q. H.; Recht, B.; Kanazawa, A. Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5491–5500, 2022.

[45] Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics* Vol. 41, No. 4, Article No. 102, 2022.

[46] Peng, S. D.; Zhang, Y. Q.; Xu, Y. H.; Wang, Q. Q.; Shuai, Q.; Bao, H. J.; Zhou, X. W. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9050–9059, 2021.

[47] Deng, K. L.; Liu, A.; Zhu, J. Y.; Ramanan, D. Depth-supervised NeRF: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12872–12881, 2022.

[48] Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S. M.; Barron, J. T.; Dosovitskiy, A.; Duckworth, D. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7206–7215, 2021.

[49] Abdel-Aziz, Y. I.; Karara, H. M.; Hauck, M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric Engineering & Remote Sensing* Vol. 81, No. 2, 103–107, 2015.

[50] Chen, L. C.; Zhu, Y. K.; Papandreou, G.; Schroff, F.; Adam, H. Encoder–decoder with atrous separable convolution for semantic image segmentation. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11211*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 833–851, 2018.

[51] Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In: *Field and Service Robotics. Springer Proceedings in Advanced Robotics, Vol. 5*. Hutter, M.; Siegwart, R. Eds. Springer Cham, 621–635, 2018.

Yanran Dai received her B.S. degree from the School of Communication Engineering, South-Central University for Nationalities, Wuhan, China, in 2017, and her M.S. degree from the School of Communication Engineering, Xidian University, Xi'an, China, in 2020. She is currently pursuing a Ph.D. degree in the School of Communication Engineering, Xidian University. Her research interests include camera array computational imaging and light field analysis.

Jing Li is a professor and leader of the Intelligent Signal Processing and Pattern Recognition Laboratory in the School of Telecommunications Engineering, Xidian University. She received her Ph.D. degree in control theory and engineering from Northwestern Polytechnical University, Xi'an, China, in 2008. She visited the University of Delaware, USA, from 2013 to 2014, was a research assistant in the Department of Computing, Hong Kong Polytechnic University in 2008, and a visiting scholar at the National Laboratory of Pattern Recognition, Beijing from 2004 to 2005. Her research interests include free-viewpoint video, image synthesis, and video content analysis and understanding.

Yuqi Jiang received his B.S. degree from the School of Communication Engineering, Hangzhou Dianzi University, China, in 2019. He is currently pursuing a Ph.D. degree with the School of Telecommunications Engineering, Xidian University. His research interests include array computational imaging and three-dimensional reconstruction.

Haidong Qin received his B.S. degree from Northwestern Polytechnical University in 2019. He is working towards a Ph.D. degree in the School of Computer Science, Northwestern Polytechnical University. His research interests include free-viewpoint video.

Bang Liang received his B.S. degree from the School of Information and Communication, Guilin University of Electronic Technology, China, in 2018 and his M.S. degree from the School of Computer Science, Northwestern Polytechnical University in 2021. His research interests include three-dimensional reconstruction and free-viewpoint synthesis.

Shikuan Hong received his B.S. degree in electronic information engineering from Hebei University of Engineering, China, in 2019. He is currently pursuing an M.S. degree in the School of Communication Engineering, Xidian University. His research interests include multiple camera arrays and natural image registration.

Haozhe Pan received his B.E. degree from the School of Communication Engineering, Wuhan University of Technology, China, in 2020. He is currently pursuing an M.S. degree in the School of Communication Engineering, Xidian University. His research interests concentrate on three-dimensional reconstruction.

Tao Yang is a professor in the School of Computer Science, Northwestern Polytechnical University, where he received his Ph.D. degree in control theory and engineering in 2008. He was a visiting scholar at the University of Delaware, USA, from 2013 to 2014, a postdoctoral fellow at Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, Northwestern Polytechnical University from 2008 to 2010, a research intern at FX Palo Alto Laboratory, CA, USA from 2006 to 2007, and a visiting scholar of the Intelligent Video Surveillance Group, National Laboratory of Pattern Recognition, Beijing from 2004 to 2005. His research interests include camera array computational imaging and 3D vision.