

A Noncoverage Field Model for Improving the Rendering Quality of Virtual Views

Changjian Zhu, Li Yu [✉], *Member, IEEE*, and Zixiang Xiong, *Fellow, IEEE*

Abstract—Rendering quality optimization theory is one of the most basic and fascinating components of image-based rendering (IBR). The rendering quality of virtual views is related to the information in the source images. Capturing the image information depends on the geometric configuration of the camera (GCC) and, in particular, the positions and shooting directions of the cameras. Therefore, the rendering quality of virtual views can be improved by optimizing the GCC. This paper investigates the relationship between the GCC and the geometric configurations of the virtual view (GCVV). The influence of the GCC on the rendering quality is also analyzed. Based on the relationships among the GCC, GCVV, and rendering quality, a mathematical model of the noncoverage field (NCF) is proposed to quantify the rendering quality. The performance of the NCF is also analyzed using a set of GCVVs and GCCs. Furthermore, an optimization algorithm based on the NCF that simultaneously optimizes the position and direction of the GCC is presented. The proposed technique can be applied to obtain the optimal rendering quality of IBR for the linear case of camera positions and virtual views. Finally, experimental results are presented and compared with the theoretical results.

Index Terms—Image-based rendering, virtual view, camera, geometric configuration, rendering quality assessment.

I. INTRODUCTION

IMAGE-BASED rendering (IBR) techniques can synthesize arbitrary novel views of a 3D scene from certain 2D multi-view images, which are captured by cameras [1], [2]. IBR is now widely used in diverse applications, such as immersive communication [3], 3D television (3DTV) [4] and free-viewpoint TV (FTV) [5]–[7] systems. Generally, these systems are composed

of an acquisition component through which the image information is captured using a given geometric configuration of the camera (GCC, i.e., the position and shooting direction) and a component that renders novel views [8]–[11]. In particular, the acquisition component is very important to light field (LF) based methods, which do not use in-depth information for auxiliary mapping [12]–[15]. The LF approach is the most well-known modern view synthesis method related to the IBR for achieving high quality video systems. In this work, we are mostly interested in the optimization acquisition component.

For the acquisition component, the minimum number of multi-view images necessary for alias-free rendering can be approximately determined using plenoptic sampling theory [16]–[19]. The success of plenoptic sampling theory depends on the assumption of a band-limited light field signal [20], [21], which is difficult to obtain for complex objects that exhibit occlusion and non-Lambertian reflection. Therefore, in most real-world scenarios, alias-free rendering is challenging, even from the point of view of plenoptic sampling theory.

Furthermore, image acquisition typically employs many cameras simultaneously to perform the capturing task [22], [23]. Generally, the total number of cameras is fixed for the duration of acquisition in a practical FTV system (or immersive communication or 3DTV system) [24]. Additionally, the image information that is captured is closely related to the GCC [25]. For example, the acquisition information for a scene can be changed using different camera positions and directions. Thus, the quality of the reconstruction is also related to the position and direction of the cameras used to capture the images. Capturing a set of multi-view images of a scene is a fundamental issue; therefore, to improve the rendering quality of the virtual views, simultaneous optimization of the positions and directions that compose the GCCs is an important research topic for IBR.

Our work is inspired by the results of the above analysis. We first analyze rendering quality based on whether it can be improved by simultaneously optimizing the position and direction of the GCC. Then, to optimize the GCC dynamically, a mathematical model, called the non-coverage field (NCF), is proposed to quantify the relationship between the rendering quality and the GCC. We give an equation to express the rendering quality based on the NCF area, and the rendering quality of virtual views is assessed using the NCF. Then, the NCF is used to devise an algorithm to optimize the GCC to improve the rendering quality. The optimization algorithm is based on modified fuzzy c-means (MFCM) clustering. Our framework builds upon Zhang *et al.*

Manuscript received June 1, 2016; revised December 29, 2016 and April 23, 2017; accepted August 5, 2017. Date of publication September 15, 2017; date of current version February 14, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61231010, and in part by the 863 High-Tech Research and Development Program under Grant 2015AA015903. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Maria G. Martini. (*Corresponding author: Li Yu.*)

C. Zhu and L. Yu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: changjianzhu@hust.edu.cn; hustlyu@hust.edu.cn).

Z. Xiong is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77840 USA, and also with the Department of Electrical and Computer Systems Engineering, Monash University, Clayton, VIC 3168, Australia (e-mail: zx@ece.tamu.edu).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org> or at http://ei.hust.edu.cn/lab/mcnc-lab/index.html#/resource/other/TMM_NCF_paper_supplemental_material.pdf provided by the author. The material includes TMM_NCF_paper_supplemental_material.pdf. Contact hustlyu@hust.edu.cn for further questions about this work.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMM.2017.2750409

[22] and extends it in several aspects, such as optimizing the camera direction for unstructured light fields [26].

Our proposed scheme provides three contributions to the IBR research field:

- 1) The NCF model is simple to use for describing variations in the GCC. It can also be applied to quantify the rendering quality. Thus, the NCF model can greatly simplify the calculation complexity of GCC optimization theory.
- 2) In the MFCM algorithm, a minimal amount of geometric information is needed because the NCF model depends only on the coverage area between the camera and virtual view.
- 3) The position and direction of the cameras are simultaneously optimized using the MFCM algorithm. Given the total number of available cameras, the GCC can be optimized using the proposed algorithm, thereby improving the rendering quality.

The remainder of this paper is organized as follows. Related works are reviewed in Section II. Standard parameterizations, the definition of our NCF model are presented in Section III. Analysis of rendering quality quantification using NCF is presented in Section IV, and then the MFCM algorithm is described. Section V presents experimental results. Conclusions are provided in Section VI.

II. RELATED WORK

A. Existing Methods for Quantifying the Rendering Quality

Regarding quantification of the rendering quality, Nguyen *et al.* [27] noted that the IBR quality can be quantified using IBR parameters, such as the depth and intensity estimate errors, the number of actual cameras, and their positions and resolutions. They used the per-pixel depth to quantitatively analyze the rendering quality of IBR. Liu *et al.* [28] applied the proper number of images for rendering and a 3D surface to describe the relationship between multi-view data capturing and the quality of the rendered view. Schirmacher *et al.* [29] presented a method for the adaptive acquisition of images for Lumigraphs from synthetic scenes. The authors used image warping to predict the potential improvement in image quality when adding a certain virtual view. Their method enables the Lumigraph creator to specify the desired number of viewpoints and then automatically construct a Lumigraph. Shidanshidi *et al.* [30] proposed a quantitative approach for comparing and evaluating of light field rendering algorithms to overcome the limitation of light field rendering methods that provide only a subjective visualized output. This approach was called the effective sampling density of a scene. Fleishman *et al.* [31] defined a quality measure using the surface observed, or covered, by the given viewing area. This quality measure can be applied to address the problem of camera placement when covering a given 3D model.

In particular, we focus on GCC optimization, and neither the camera positions nor their directions are restricted for unstructured light field rendering. Therefore, in this report, we analyze the relationships between the camera positions and the rendering quality and between the camera directions and the rendering quality. Nevertheless, the position and direction of the GCC are

two different measurements. We thus employ the NCF model to describe variations in the GCC. Similar to the NCF model, Buehler *et al.* [26] has proposed a camera blending field that can be applied to measure the unstructured input of cameras for the rendering weights of a light ray, the weights that contain the angle between the cameras and the rendering light ray, the resolution of the cameras and the field-of-view of the cameras. Safaei *et al.* in [24] proposed a correspondence field, which is a mathematical representation of a multi-camera geometry, to quantify the suitability of a camera configuration for a given arrangement of objects.

B. Existing Algorithms for Optimizing the GCC

The goal of such algorithms is to improve the rendering quality. For instance, Schirmacher *et al.* [29] proposed an algorithm for adaptive capturing of images from synthetic scenes using Lumigraphs. They concluded that new virtual views of the scene should be rendered and added to the light field based on the image quality. Fleishman *et al.* [31] proposed an automatic camera placement algorithm that addresses the problem of placing cameras in a manner that covers a given 3D model. Their goal was to place the cameras in a location that optimizes the texture map formed by the captured images for a known mesh model. Their method first determines an approximation for the set of surfaces that are visible from a given viewing area and then selects a small set of appropriate camera positions to capture the scene. In [32], Werner *et al.* applied the growth and selection of plausible viewing intervals to find the suboptimum objective function via heuristic optimization. Their research sought to select an optimal set of reference views. The optimal set was defined as that which minimizes a given objective function. Zhang *et al.* [22] presented an active rearranged capturing (ARC) algorithm to prove that the rendering quality can be improved by optimizing the camera positions on a plane. These research results were primarily based on the assumption that the camera directions are constant and can be fixed a priori; in this case, only the camera positions can vary. The authors assumed that given a total number of available cameras, ARC moved the cameras strategically on the camera plane until the sum of the squared rendering errors was minimized for a given set of light rays requiring rendering. They found that moving the cameras on a set of rails allowed them to perform ARC and improve the rendering quality rapidly.

Recently, a number of research efforts [33]–[37] have focused on unstructured Lumigraph rendering and unstructured light field rendering. These methods allow for Lumigraph/light field-style rendering from a set of input cameras that are not restricted to a plane or to any specific manifold. Several recent works attempted to find the optimal GCC of unstructured light field rendering. For example, Buehler *et al.* [26] proposed an unstructured Lumigraph rendering algorithm that directly rendered virtual views from an unstructured collection of input images, associated with the camera pose estimates, to obtain a real-time rendering protocol. Snavely *et al.* [38] studied the optimal unstructured camera path based on the results of [26]. They used a large set of community and personal photos to

TABLE I
NOTATIONS AND DEFINITIONS

| | |
|---|--|
| (x_0, y_0) | The camera position in a plane |
| (x_1, y_1) | The virtual view position in a plane |
| θ_c | The rotation angle of the camera |
| θ_v | The rotation angle of the virtual view |
| $C_n(x_0, y_0, \theta_c)$ | The geometric configuration of the n th camera |
| $V_m(x_1, y_1, \theta_v)$ | The geometric configuration of the m th virtual view |
| $\hat{C}_n(\hat{x}_0, \hat{y}_0, \hat{\theta}_c)$ | The optimal GCC |
| u | The pixel of the image plane of the camera |
| l_i | The i th light ray of the camera |
| l_j | The j th desired light ray of the virtual view |
| u_i | The i th pixel point for the image plane of camera |
| v_j | The j th desired pixel point of the virtual view |
| Δv_j | The rendering error of the j th desired light ray |
| e_m | The rendering error of the m th virtual view |
| Λ_c | The volume of field-of-view of a camera |
| Λ_v | The volume of field-of-view of a virtual view |
| Λ_{nm} | The volume of a NCF between C_n and V_m |
| Λ_{CF} | The volume of a CF between C_n and V_m |
| S_{nm} | The area of a NCF between C_n and V_m |
| η_{nm} | The membership function between C_n and V_m |
| Z_c | The constant depth, which limits the NCF area |

reconstruct virtual views. The authors automatically computed the capturing orbits of the cameras using panoramas and canonical views to obtain the optimal paths between the cameras. Davis *et al.* [39] studied unstructured light fields. The researchers relied on a hand-held camera and laptop to capture the images and implement the IBR system. They applied the latest simultaneous localization and mapping technology to measure the area from which the hand-held cameras had the maximum coverage of the scene. In [40] and [41], Shidanshidi *et al.* used the effective sampling density to establish an optimization model that determines the optimal configurations of the capturing cameras, including their positions and directions.

III. STANDARD PARAMETERIZATION AND DEFINITIONS

For clarity, we list the important notations and the associated definitions used throughout the paper in Table I.

A. Problem Statement

We define an ideal pinhole camera following the camera model introduced by Tsai [42] as shown in Fig. 1. In this model, the camera is modeled by a six-dimensional function $C(x, y, z, \theta_x, \theta_y, \theta_z)$, which describes the camera view at position (x, y, z) when looking in the direction $(\theta_x, \theta_y, \theta_z)$. The parameters of the camera model are referred to as the GCC. Similarly, the parameters $V(x, y, z, \theta_x, \theta_y, \theta_z)$ of the virtual view are referred to as the geometric configuration of the virtual view (GCVV). As shown in Fig. 2, we assume that N cameras are used to capture images in a static scene. M virtual views are rendered by the captured images for IBR. Both N and M are positive finite integers, with $N < M$. In IBR, for several reasons, such as insufficient sampling and inaccurate geometric modeling, the rendering quality of virtual views is not perfect [22]. In this study, the only errors are assumed to arise from inaccurate camera positions and directions. The rendering error of a virtual view is e_m , which is the difference between the

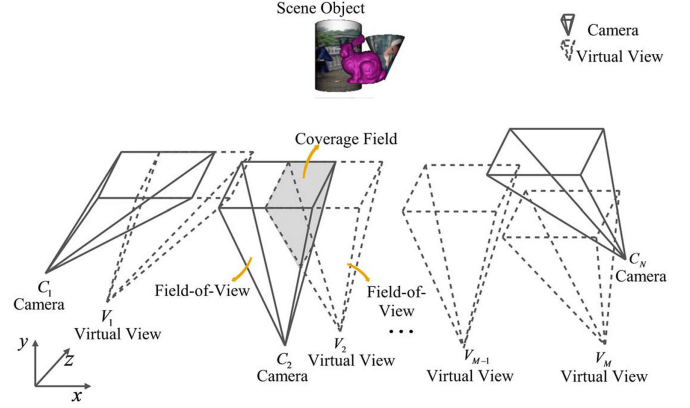


Fig. 1. Diagram showing an ideal pinhole camera model and virtual view model. This diagram also shows the definition of a CF model that is between a camera and virtual view, where the coverage range of the shadow is a CF model in 3D space.

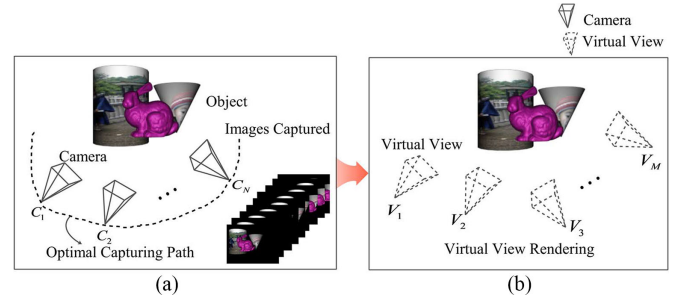


Fig. 2. Diagram of the IBR method. (a) A set of images is captured, when the optimal capturing path is determined. (b) A number of virtual views are rendered from which the captured images.

rendered virtual view and the actual view. Below, the rendering error is applied to describe the rendering quality. For all cases, n is the index of the camera and m is the index of the virtual view, where $n = 1, 2, \dots, N$ and $m = 1, 2, \dots, M$.

As shown above, the rendering error is related to the GCC and GCVV. We are primarily interested in finding the best methods to simultaneously translate (x, y, z) and rotate $(\theta_x, \theta_y, \theta_z)$, in the case where the GCVV is fixed in advance, to improve the rendering quality of virtual views. We define the GCC optimization problem as follows.

Definition 1: Given M virtual views to be rendered by N cameras, the optimal geometric configuration of the N cameras is defined as

$$\hat{\mathbf{C}}(\hat{x}, \hat{y}, \hat{z}, \hat{\theta}_x, \hat{\theta}_y, \hat{\theta}_z) = \arg \min_{\mathbf{C}(x, y, z, \theta_x, \theta_y, \theta_z)} \Psi(\mathbf{e}), \quad (1)$$

where $\Psi(\mathbf{e})$ is a function of the rendering error of the virtual views, depending on the particular application [22]. Moreover, $\hat{\mathbf{C}} = \{\hat{C}_n, n = 1, 2, \dots, N\}$, $\mathbf{C} = \{C_n, n = 1, 2, \dots, N\}$. In this study, $\Psi(\mathbf{e})$ is defined as

$$\Psi(\mathbf{e}) = \sum_{m=1}^M (e_m)^2. \quad (2)$$

B. NCF Model Definition

Assume that a camera and a virtual view can be freely moved and rotated in a limited angle range. As shown in Fig. 1, the dotted line denotes the coverage range of the virtual view, and the volume of the coverage range is represented by Λ_v . We consider that the scene object is completely included in the coverage range of each virtual view. Similarly, the solid line is the coverage range of the camera, and its volume is represented by Λ_c . The coverage range of the shadow is a topology of the intersecting coverage range of the field-of-view between the camera and the virtual view. We refer to the topology of the intersecting coverage range as the coverage field (CF). Intuitively, the topology of the CF in Fig. 1 is a cube. Its parameterization mainly depends on the points of intersection between the GCC and GCVV. Without the loss of generality, the volume of a CF is expressed as

$$\begin{aligned}\Lambda_{CF} &= \Phi_V(x, y, z, \theta_x, \theta_y, \theta_z) \\ &= \iiint_{CF} \Phi_f(x, y, z, \theta_x, \theta_y, \theta_z) dx dy dz, (x, y, z) \in CF,\end{aligned}\quad (3)$$

where $\Phi_V(\cdot)$ is the volume function of a CF between a camera and a virtual view, and $\Phi_f(\cdot)$ is a function of the cube vertices.

Because both the GCC and GCVV are randomly distributed, the variation of the CF with the GCC and GCVV is also random. In this case, the relationship between the GCC and GCVV can not be described only by the CF. To solve this problem, as shown in Fig. 1, we defined another model of NCF Λ_{nm} to represent the volume of the area that is not mutually covered between the camera and the virtual view. Obviously, $\Lambda_{nm} = \Lambda_c + \Lambda_v - 2\Lambda_{CF}$.

IV. OPTIMIZATION ALGORITHM FOR THE GCC

In this section, we introduce a GCC optimization algorithm based on the NCF model. Our goal is to improve the rendering quality of M virtual views by optimizing the geometric configuration of N cameras.

A. A Optimization Model Using the NCF

Without loss of generality, the NCF Λ_{nm} can be applied to describe the variation of GCC, with the end goal of using the NCF to optimize the GCC. We now state the GCC optimization method using Λ_{nm} in the following theorem.

Theorem 1: When the NCF Λ_{nm} is minimal, the GCC is optimal, and the resulting optimal GCC can be represented as a function of the NCF Λ_{nm} as

$$\hat{\mathbf{C}}(\hat{x}, \hat{y}, \hat{z}, \hat{\theta}_x, \hat{\theta}_y, \hat{\theta}_z) = \arg \min_{\mathbf{C}(x, y, z, \theta_x, \theta_y, \theta_z)} \sum_{m=1}^M \kappa_m \min_{n=1, 2, \dots, N} \Lambda_{nm}, \quad (4)$$

where κ_m is a scaling factor.

Proof: The full proof can be found in Appendix A. A brief outline of the proof is as follows. As can be observed in Fig. 1, the contribution of the virtual view to the camera's rendering

increases with decreasing Λ_{nm} . Intuitively, we find that when the GCC changes, the variation of Λ_{nm} is the same as the rendering error of the virtual view. That is, as Λ_{nm} decreases, the rendering error decreases. We consider that there is a linear approximation between Λ_{nm} and the rendering error. Based on a very coarse approximation, we can assume

$$e_m = \Phi_m(\Lambda_{nm}) \approx \kappa_m \min_{n=1, \dots, N} \Lambda_{nm}. \quad (5)$$

Substituting (5) into (1), we solve the above optimization problem and obtain the optimal GCC as given in (4). ■

B. GCC Optimization Based on a Simplified NCF Model

In (4), to apply the NCF model to perform the GCC optimization tasks, the rendering quality and NCF model must be parameterized. Nevertheless, it can be observed from (1), (3) and (4) that addressing the six dimensions of the camera model is not trivial, and thus the cube topology of the CF cannot be simply parameterized. The reason is that the topology of the CF is an irregular cube, and the derivation of (3) depends on the shape of the irregular cube. Thus, simplifications of the camera and the NCF model are first presented to establish an algorithm that rapidly optimizes the GCC.

1) Simplification of the Optimization Problem: As mentioned the above, optimizing the GCC is a complex task when considering all six dimensions simultaneously. Fortunately, the influence weights of the six dimensions on the captured information of the multi-view images are different. Additionally, the coordinates of the camera can be described by two primary components for the position and direction. Inspired by the unstructured light field method [26], two dimensions of the spatial position and another one of the spatial direction can typically describe the primary variations of the GCC. Therefore, we can select the dimensions of the camera position (x, y) and the viewing direction θ_y in our study. Thus, (z, θ_x, θ_z) remains fixed, and only the (x, y, θ_y) dimensions are varied. In this report, we define two variables (x_0, y_0) to represent the camera position (x, y) , and define a variable θ_c to represent the camera direction θ_y . Therefore, the n th camera is described by $C_n(x_0, y_0, \theta_c)$. Similarly, the function $V_m(x_1, y_1, \theta_v)$ is used to describe the virtual view that is observed at position (x_1, y_1) when looking in direction θ_v . As shown above, Fig. 2 can be simplified to Fig. 3, i.e., the six dimensions of the GCC optimization are simplified to the three dimensions, x_0, y_0 and θ_c . In Fig. 3, the cameras and virtual views can be freely moved along a plane and rotated in a limited angular range. Then, (1) is simplified to

$$\hat{\mathbf{C}}(\hat{x}_0, \hat{y}_0, \hat{\theta}_c) = \arg \min_{\mathbf{C}(x_0, y_0, \theta_c)} \Psi(\mathbf{e}). \quad (6)$$

Based on this simplified method, the rendering quality and NCF model are parameterized to study the GCC optimization.

2) Simplification of the NCF Model: The field-of-view of the camera in Fig. 1 can be regarded as a tetragonal cone, and each tangent plane through the vertex of the tetragonal cone is a triangle. To simplify the parameterization of the NCF,

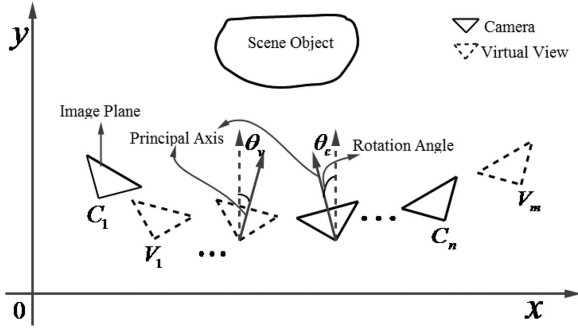


Fig. 3. Diagram of the IBR in 2D space. This diagram also shows the images that are captured and the virtual views that are rendered.

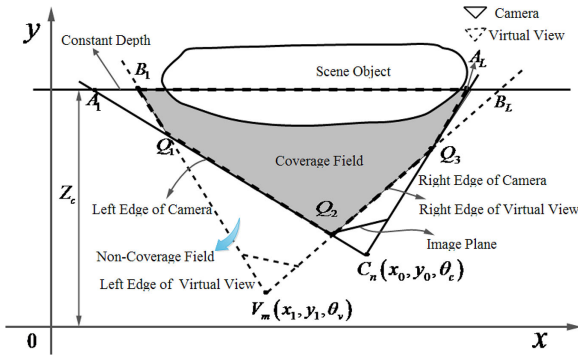


Fig. 4. Diagram showing the definitions of the NCF and CF that are between a camera and a virtual view, where the coverage range of the shadow is a CF in the 2D plane. This diagram also shows the parameterization of the NCF area. The constant depth line $y = Z_c$ limits the NCF area; therefore, only the NCF area for $y \leq Z_c$ is calculated. Then, depending on the changes in the GCC and GCVV, the CF topology of the bold black dotted line may be a triangle, square, or pentagon. The CF is parameterized using the area of a polygon, $\odot B_1 Q_1 Q_2 Q_3 A_L$.

similar to the analysis presented in (6), we select only one tangent plane of the NCF in Fig. 1 for NCF parameterization. Therefore, the cube topology of the CF is simplified to a 2D plane topology, as shown in Fig. 4. This figure shows the top view of the plane that is associated with one row of pixels. For the sake of clarity, only two light rays of the field-of-view on the edge are shown in the plane. The coverage range of either the virtual view or the camera is the area between the left and right edges. The area of four triangles for the NCF is not mutually covered by the camera and the virtual view. These triangles are $\Delta A_1 Q_1 B_1$, $\Delta Q_1 V_m Q_2$, $\Delta Q_2 C_n Q_3$ and $\Delta Q_3 B_L A_L$. Using the variable S_{nm} to represent the area of the four triangles, $S_{nm} = S_v + S_c - 2S_{CF}$, where S_v is the area of $\Delta B_1 V_m B_L$ and S_c is the area of $\Delta A_1 C_n A_L$, these areas can be calculated. Similar to Λ_{nm} , S_{nm} can also be used to describe the topology of the intersecting coverage range of the field-of-view between the camera and the virtual view, thus the relationship between the GCC and the GCVV can be described by the NCF area. Additionally, the volume calculation of the NCF is simplified to a polygon area calculation. Intuitively, the NCF volume Λ_{nm} decreases as the S_{nm} decreases.

3) *Light Ray Parameterization*: The image information space is expressed by a set of light rays [20], [21]. Given a light

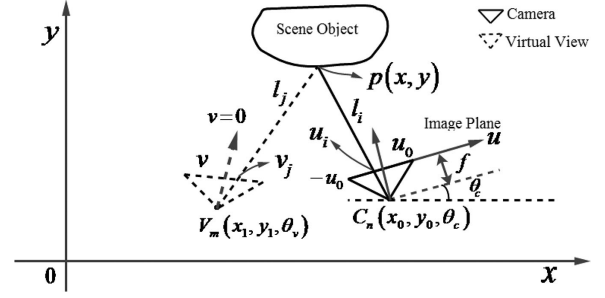


Fig. 5. This diagram shows the parameterization of the rendering error [8]. u_i is the i th captured pixel point. The desired light ray l_j intersects the image plane of the virtual view at the desired pixel point v_j . The desired light ray l_j is rendered by the light ray l_i . If any difference between the GCC and GCVV exists, errors will occur in the rendering result for l_j . The rendering error Δv_j of l_j with respect to l_i is defined as $|u_i - v_j|$.

field rendering, one can generate a new virtual view from an arbitrary captured image using light ray interpolation. To facilitate the parameterization of the rendering error and NCF area, the light rays are parameterized using homogeneous coordinates based on the position and direction axes following Hartley and Zisserman [43]. As shown in Fig. 5, the initial direction of the camera is the vertical direction of the x -axis, which indicates that the camera is not rotated. The light ray parameterization defines each light ray by its intersection with two parallel planes, the camera plane (x_0, y_0) , and the image plane u in the 2D space. The separation between the two planes is equal to the focal length f . An epipolar plane image (EPI) [44] of the 2D light field can be constructed if we plot its intensity at all possible (x_0, y_0, u) coordinates. Moreover, if $\theta_c > 0$, the camera is rotated clockwise. If $\theta_c < 0$, the camera is rotated counter-clockwise. Finally, if $\theta_c = 0$, the camera is not rotated. Note that we consider a finite field-of-view, hence $-u_0 \leq u \leq u_0$, and we define the finite field-of-view of a camera consisting of L light rays in a 2D space. The first light ray is at the left edge of the field-of-view, and l_i is the i th light ray of the camera, where $i = 1, 2, \dots, L$. The pixel point u_i is the point for light ray l_i that intersects the image plane. As shown in Fig. 5, the light ray l_i of the camera is

$$y = y_0 + \frac{(f \cos \theta_c + u_i \sin \theta_c)(x - x_0)}{u_i \cos \theta_c - f \sin \theta_c}. \quad (7)$$

Similarly, the desired light ray can also be expressed via (7). We define the variable l_j as the j th desired light ray for a virtual view, where $j = 1, 2, \dots, L$. The desired light ray l_j of a virtual view can be expressed as

$$y = y_1 + \frac{(f \cos \theta_v + v_j \sin \theta_v)(x - x_1)}{v_j \cos \theta_v - f \sin \theta_v}. \quad (8)$$

Point v_j is referred to as the desired pixel point, which is the point where l_j intersects the image plane of the virtual view.

4) *Rendering Quality Quantification Using the NCF Area*: Based on the above simplification of the GCC optimization problem and NCF model, the quantification of NCF area to quantify the rendering error is given in the following theorem.

Theorem 2: When the GCC is changed, the tendency of change in the rendering error is the same as the tendency of

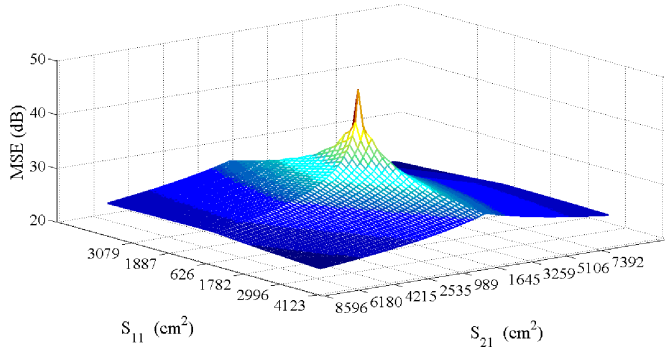


Fig. 6. The rendering error with respect to the NCF area. The MSEs and NCF areas are calculated with both a camera position spacing of 1.0 cm and a rotation angle spacing of 0.1 rad according to (9). The virtual view's position is (0.0, 0.0) cm, and the rotation angle is 0.0 rad.

change in the NCF area. Thus, there is an approximately linear relationship between the NCF area and the rendering error. It can be expressed as

$$e_m = \Phi_{\Delta v}(S_{nm}) \approx \omega_m \min_{n=1, \dots, N} S_{nm}, \quad (9)$$

where ω_m is a scaling factor.

Proof: The full proof can be found in Appendix B. A brief outline of the proof is as follows. First, we use the mean square error (MSE) of the desired light rays to calculate the rendering error of the virtual view as

$$e_m = \Phi_{\Delta}(x_0, y_0, \theta_c) = \sqrt{\frac{1}{L} \cdot \sum_{j=1}^L (\Delta v_j)^2}, \quad (10)$$

where Δv_j represents the rendering error of l_j with respect to l_i as Fig. 5. Subsequently, as shown in Fig. 4, the area of the CF is calculated using the intersection points, A_1, A_L, B_1 and B_L . Then, the area of the NCF is expressed as

$$S_{nm} = \Phi_s(x_0, y_0, \theta_c, x_1, y_1, \theta_v). \quad (11)$$

Additionally, from (10) and (11), the values of e_m and S_{nm} depend on x_1, y_1, θ_v, x_0 , and θ_c . As the difference between (x_1, y_1, θ_v) and (x_0, y_0, θ_c) decreases, the value of e_m also decreases and S_{nm} decreases. Finally, by (11), the camera position x_0 can be represented using S_{nm} from the CF area function $S_{nm} = \Phi_s(x_0, y_0, \theta_c, x_1, y_1, \theta_v)$. It can be written as

$$x_0 = \Phi_t(S_{nm}) = \begin{cases} t_I, & \text{if } x_1 < x_0 < x_1 + W \\ t_{II}, & \text{if } x_1 - W < x_0 \leq x_1 \end{cases}, \quad (12)$$

where $W = 2Z_c u_0 / f$ and t_I and t_{II} are the camera positions which are calculated by the NCF area, as shown in Appendix B. Thus, substituting (12) into (10), the rendering error function $\Phi_{\Delta v}(\cdot)$, which is calculated using the NCF area can be represented using (9). Therefore, the rendering error of the virtual view can be quantified using the NCF area. ■

To intuitively illustrate the above conclusion, we present an example of (9), which is illustrated in Fig. 6. In this example, the camera position and direction are simultaneously changed

in terms of the MSE and NCF area. It should be noted that as the NCF area from the camera to the virtual view decreases, the mean squared error (MSE) decreases. Three examples are shown in Fig. 20 in Appendix C, in which the influences of the GCC on the rendering error and NCF area are presented using (10) and (11). From Figs. 6 and 20, when the GCC is changed, as the NCF area decreases, the rendering error decreases.

Subsequently, the effects of the camera position and direction on the rendering quality and the NCF area are also analyzed using three experiments. The full results of the experiments are provided in Appendix C, which is presented in the supplementary material. A brief outline of the explanation is as follows. The synthetic experiment results in Figs. 21–23 yielded a relationship between the rendering quality and NCF area that is similar to that obtained by the theoretical analysis results shown in Figs. 6 and 20. The rendering error and NCF area both decrease as either the distance between the positions of the camera and the virtual view decreases or as the angular difference between the directions of the camera and the virtual view decreases.

Based on the above analysis, it can be observed that when the NCF area sufficiently describes the GCC, the rendering quality can be quantified using the NCF area. Therefore, the NCF area can be used to establish an algorithm that optimizes the GCC to improve the rendering quality.

5) A Optimization Algorithm Using the NCF Area: Because of theorem 1 presented in Section IV-A, the optimal GCCs with M virtual views can be solved using Λ_{nm} according to (4). However, as previously mentioned, the calculation of Λ_{nm} is too complicated, thus there is no need to apply the Λ_{nm} to optimize the GCCs. Additionally, the cube topology of the CF can be simplified to a 2D plane topology as defined in Section IV-B. In this case, Λ_{nm} decreases as the NCF area decreases. Thus, in this sub-section, we use the NCF area instead of Λ_{nm} in (4) to establish an algorithm that optimizes the GCCs. In addition, as defined previously, the NCF area is associated with the virtual view $V_m(x_1, y_1, \theta_v)$ and camera $C_n(x_0, y_0, \theta_c)$. For each virtual view, there are N NCF areas; however, we select only the minimum NCF area to serve as the NCF of the virtual view. Therefore, these M virtual views can be divided into N groups according to the area of the corresponding NCF; the number of virtual views for the n th group is M_n . From Appendix C, the relationship between the rendering error and the multiple NCFs can be described as the sum of multiple NCF areas. Therefore, combining (4), (9) and (11), (6) can be rewritten as a function of the NCF area,

$$\hat{\mathbf{C}}(\hat{x}_0, \hat{y}_0, \hat{\theta}_c) = \arg \min_{\mathbf{C}(x, y, \theta_c)} \sum_{m=1}^M \left((\eta_{nm})^b \min_{n=1, \dots, N} |S_{nm}| \right), \quad (13)$$

where $n = 1, \dots, N$, $m = 1, \dots, M$, and η_{nm} is a membership function between $C_n(x_0, y_0, \theta_c)$ and $V_m(x_1, y_1, \theta_v)$.

We now present a modified fuzzy c-means (MFCM) clustering algorithm [45] to solve (13). Fuzzy c-means (FCM) is very powerful method for solving certain types of clustering problems with a population Γ of τ objects. A cluster is represented by a vector of γ numerical features or measurements $\psi_m \in \mathbb{R}$. The cluster must be decomposed into sub-populations

(or clusters) of similar objects, where \mathbb{R} is a set of real numbers. The c-means objective function J_b is defined by the Euclidean distance $d_{nm} = \|\psi_m - \xi_n\|^2$, where $\{\psi_m\}$ is an iterative sequence, b is a weighted index, and ξ_n is the cluster center [45]. Based on the FCM algorithm, we define the objective function $J_b(U, X_c, R)$ based on the NCF area as

$$J_b(U, X_c, R) = \sum_{n=1}^N \sum_{m=1}^{M_n} \left((\eta_{nm})^b \cdot S_{nm} \right), \quad (14)$$

where $U = \{\eta_{nm} | n = 1, \dots, N, m = 1, \dots, M_n\}$ is the membership function set. Noted that the membership function is calculated using the MFCM method. It requires that the sum of the membership functions of every cluster for a sample equals 1, i.e.,

$$\sum_{n=1}^N \eta_{nm} = 1, \forall m = 1, 2, \dots, M. \quad (15)$$

Additionally, the GCC generally varies within a limited boundary. Thus, let $X_v = \{x_1, y_1\}$ be bounded in \mathbb{R} , let $X_c = \{x_0, y_0\}$ be the camera position set and let $R = \{\theta_c\}$ be the rotation angle set of the camera, with $x_0, y_0 \in \mathbb{R}, \theta_c \in [-\pi/2, \pi/2]$. For the weighted index, Choe *et al.* [46] noted that the performance of the FCM algorithm is not sensitive to the weighted index. However, Pal *et al.* [45] determined that the weighted index must be in the range [1.5, 2.5], and is typically set to 2.0. Therefore, b is set to 2.0 in this study.

C. Convergence of the Optimization Algorithm

Regarding the convergence of the FCM, Bezdek [47] presented an application of Zangwill's theory [48] for FCM iterations that demonstrated that the iteration sequence (or, at worst, a subsequence thereof) of the FCM converges to a local minimization of J_b [49]. We analyze the convergence of $J_b(U, X_c, R)$ with S_{nm} of the MFCM algorithm. Let x_1, y_1 and θ_v be three constants, and x_0, y_0 and θ_c denote variables. Let $x_1, y_1 \in \mathbb{R}$ and two fixed values, thus $x_0 \in [x_1 - W, x_1 + W], y_0 \in [y_1 - W, y_1 + W], \theta_v \in [-\pi/2, \pi/2]$, and $\theta_c \in [-\pi/2, \pi/2]$. From (11), S_{nm} and $J_b(U, X_c, R)$ are minimized when $x_0 = x_1, y_0 = y_1$ and $\theta_c = \theta_v$. This can also be observed from Figs. 6 and 20 in Appendix C. According to the above analysis, we can obtain the minimum of S_{nm} as

$$(S_{nm})^* = \{S_{nm}, \text{ when } x_0 = x_1, y_0 = y_1, \theta_c = \theta_v\}. \quad (16)$$

By (14) and (16), the minimum of $J_b(U, X_c, R)$ is

$$J_b(U, X_c, R)^* = \{J_b(U, X_c, R), \text{ if } S_{nm} = (S_{nm})^*\}. \quad (17)$$

Therefore, both S_{nm} and $J_b(U, X_c, R)$ are global convergence functions on x_0, y_0 and θ_c with limited sets. Convergence of the MFCM algorithm can be obtained by

$$\{J_b(U, X_c, R)^* \geq J_b(U, X_c, R), \forall \theta_c \in [-\pi/2, \pi/2], \\ x_0 \in [x_1 - W, x_1 + W], y_0 \in [y_1 - W, y_1 + W]\}, \quad (18)$$

$$\Omega = \{J_b(U^r, (X_c)^r, (R)^r)\} \rightarrow J_b(U, X_c, R)^*, r = 1, 2, \dots \quad (19)$$

where Ω is a global extremum and $W = 2Z_c u_0 / f$.

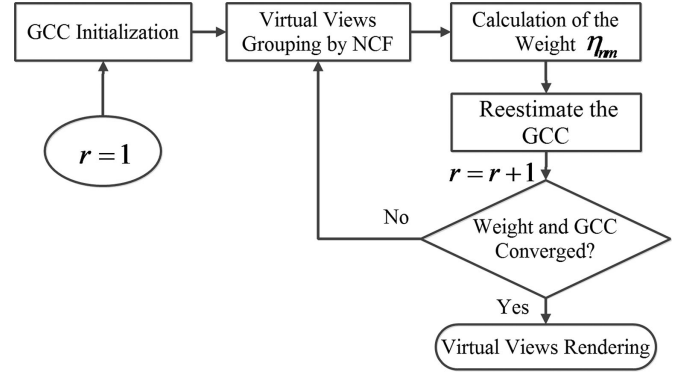


Fig. 7. The flow chart of our proposed MFCM algorithm.

From the above analysis, the minimum NCF area exists only when the GCC is the same as the GCVV. In this case, the MFCM algorithm converges, and thus the GCC is optimal.

D. The Optimization Procedure

From the above analysis, an approximate optimization of the objective function $J_b(U, X_c, R)$ based on (14) can be calculated using the MFCM algorithm. The optimization is based on iterations according to the following necessary conditions for the global extremum, and the GCC is calculated based on the MFCM algorithm. Thus, the optimization position of the camera is calculated using the GCVV under the conditions of (15) as

$$x_0 = \frac{\sum_{m=1}^{M_n} ((\eta_{nm})^b \cdot x_1)}{\sum_{m=1}^{M_n} (\eta_{nm})^b}, \quad y_0 = \frac{\sum_{m=1}^{M_n} ((\eta_{nm})^b \cdot y_1)}{\sum_{m=1}^{M_n} (\eta_{nm})^b}, \quad (20)$$

where η_{nm} is calculated as

$$\eta_{nm} = \frac{1}{\sum_{k=1}^N \left(\frac{S_{nm}}{S_{km}} \right)^{1/(b-1)}}. \quad (21)$$

Similarly, the rotation angle of the camera can be calculated using the GCVV as

$$\theta_c = \frac{\sum_{m=1}^{M_n} ((\eta_{nm})^b \cdot \theta_v)}{\sum_{m=1}^{M_n} (\eta_{nm})^b}. \quad (22)$$

Using iterative methods, (20), (21) and (22) can be solved to optimize the GCC. The algorithm illustrated in Fig. 7 is described below.

1) *GCC initialization*: Before the optimizing work, the GCC must be initialized. Generally, the GCC initialization step first provides the numbers of cameras and virtual views and the GCVV in a certain scene, i.e., it sets the values of N and M , where $M > N$. Subsequently, the GCC is uniformly or randomly initialized, as shown in Fig. 8(a).

2) *Virtual view grouping*: Based on the given GCC, the NCF area of each virtual view is calculated. Furthermore, M virtual views are divided into N groups based on the area of corresponding NCFs; the number of virtual views for the n th group is M_n .

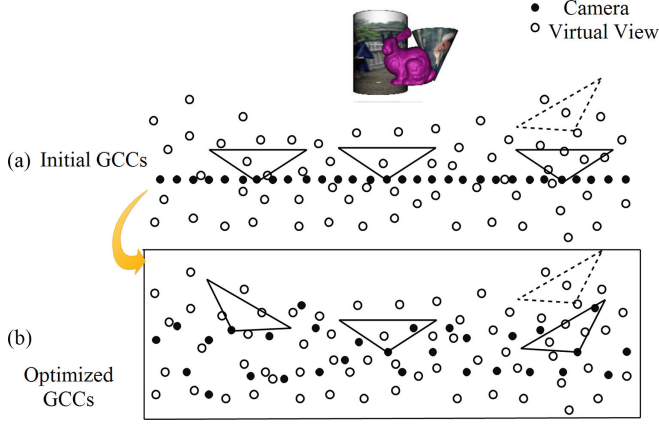


Fig. 8. Conceptual illustration of the GCC optimization procedure. The virtual views are rendered using the multi-view images captured with the optimal GCC.

3) *The membership function update*: The membership function η_{nm} is calculated using (21).

4) *The GCC re-estimated*: The membership function η_{nm} is applied to calculate the camera position x_0, y_0 and direction θ_c using (20) and (22), and then the new GCC is obtained.

5) *Convergence estimation*: If the membership degree of each virtual view is stable or the objective function $J_b(U, X_c, R)$, which is calculated using (14), is greater than a certain threshold, the optimization algorithm is converged. The optimal GCCs are obtained when the algorithm converges, as shown in Fig. 8(b). If the algorithm does not converge in the current condition of the membership function η_{nm} , then the camera position is moved and the camera direction is rotated according to the GCC re-estimation results, and then the algorithm returns to step 2).

6) *Virtual view rendering*: After the GCC is optimized, a set of optimal multi-view images is captured for the scene. Finally, the virtual views can be rendered using the captured multi-view images, and then high-quality views are obtained.

E. Generalization of the MFCM Algorithm

The previous sections were devoted to the use of the NCF area for GCC optimization. The MFCM algorithm can be used to optimize the camera positions along a plane and their directions within a limited range of angles. Generally, the NCF area can be generalized to other spatial positions and directions of cameras and virtual views, such as the z , θ_x , and θ_z axes. The NCF area can also be applied to optimize the other three variables of the GCC.

Further, when using the MFCM algorithm, the cameras and virtual views must be placed according to some restricted conditions (e.g., they are placed in a plane). However, based on (4), this algorithm can be extended to design a method for GCC optimization that is not restricted, thus improving the rendering quality of the IBR method. For example, when the virtual views are rendered from a set of input cameras that are not restricted to a 2D plane or any specific manifold, if Λ_{nm} is calculated, then the rendering quality of the virtual views can be quantified. Thus, an effective GCC optimization algorithm using Λ_{nm} can be established for IBR.

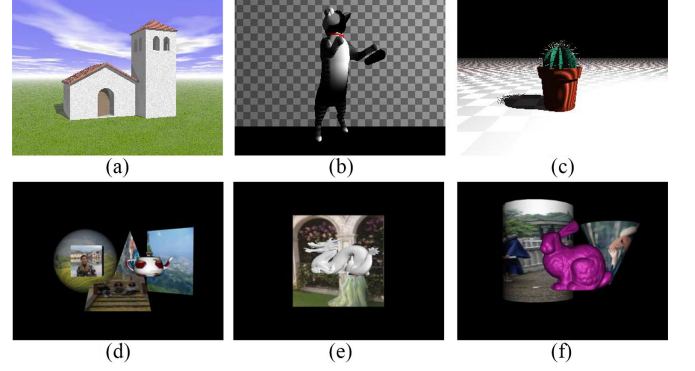


Fig. 9. (a) Church scene; (b) cat scene; (c) cactus scene. The three scenes are rendered using POV-Ray. The church, cat and cactus scenes are all fixed on the point at $(0, 100, -20)$ cm. (d) Irregular scene; (e) dragon scene; (f) bunny scene. The three scenes are rendered using 3Ds Max. The irregular, dragon and bunny scenes are all fixed on the point at $(0, 400, -40)$ cm.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, experimental results for the MFCM algorithm are presented with respect to the GCC optimization. These results are then compared with the experimental results of the ARC algorithm. For all of the experiments, the scenes and parameter settings for the camera and virtual view are the same as those in Appendix C. The reconstruction method is based on bilinear interpolation of nearby images with unstructured Lumigraph rendering [26], and little depth information is utilized.

A. The GCC Optimization Results

We collected image data sets from six synthetic scenes (see Fig. 9) to test the MFCM algorithm. In this case, 32 cameras are used to capture each synthetic scene, and then 64 virtual views are rendered using the captured images. The camera positions are uniformly initialized, and the directions are not rotated, as indicated in Fig. 8(a). The GCVVs are randomly placed, i.e., the positions of the virtual views are random within a plane, and the virtual views do not point in the same direction, as indicated in Fig. 10. Note that the GCVVs are fixed a priori, and then the GCCs are optimized using the MFCM algorithm. In this experiment, we use only a small number of cameras and virtual views. The purpose is to expediently perform the experiment and clearly present the optimization performance of the MFCM. The results for this case can be easily extended to cases with a large number of cameras and virtual views.

The MFCM algorithm converges after eight iterations in this case. The convergence time for the code running is approximately 0.051427 seconds. After optimizing the GCCs, the positions of the GCCs are non-uniform, and some of the cameras are rotated, as shown in Table II. For example, the rotation angle of the second camera is $\frac{\pi}{3}$ rad, and the rotation angle of the fifth camera is $\frac{\pi}{6}$ rad. This is because the virtual views are rotated. Hence, in this case, the optimal GCC depends on the GCVV. The optimal GCC will change if the GCVV is changed.

The rendered virtual views are evaluated using the peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) [50]. It should be observed from Fig. 11(a) that the

TABLE II
THE OPTIMAL VALUES OF THE GCCS

| n | 1, 2, 3, 4, 5, 6, 7, 8, |
|------------------------|--|
| (x_0, y_0, θ_c) | $(-97, 14, \frac{\pi}{4}), (-90, 25, \frac{\pi}{7}), (-81, 15, \frac{\pi}{8}), (-72, 20, \frac{\pi}{9}), (63, 20, \frac{\pi}{11}), (54, 30, \frac{\pi}{13}), (38, 25, \frac{\pi}{15}), (18, 10, 0),$ |
| n | 9, 10, 11, 12, 13, 14, 15, 16, 17, |
| (x_0, y_0, θ_c) | $(0, 0, 0), (18, 23, 0), (25, 0, -\frac{\pi}{13}), (34, 23, -\frac{\pi}{9}), (40, 20, -\frac{\pi}{8}), (57, 27, -\frac{\pi}{6}), (65, 20, -\frac{\pi}{5}), (80, 0, -\frac{\pi}{7}), (87, 30, -\frac{\pi}{4}),$ |
| n | 18, 19, 20, 21, 22, 23, 24, 25, |
| (x_0, y_0, θ_c) | $(100, 0, -\frac{\pi}{4}), (95, -12, -\frac{\pi}{6}), (80, -10, -\frac{\pi}{8}), (67, -23, -\frac{\pi}{10}), (50, -18, -\frac{\pi}{12}), (25, -17, -\frac{\pi}{13}), (10, -7, 0), (-20, -11, 0),$ |
| n | 26, 27, 28, 29, 30, 31, 32 |
| (x_0, y_0, θ_c) | $(-25, -27, \frac{\pi}{15}), (-37, -24, \frac{\pi}{14}), (-44, -22, \frac{\pi}{10}), (-60, -15, \frac{\pi}{7}), (-75, -30, \frac{\pi}{6}), (-82, -15, \frac{\pi}{5}), (-95, -15, \frac{\pi}{3})$ |

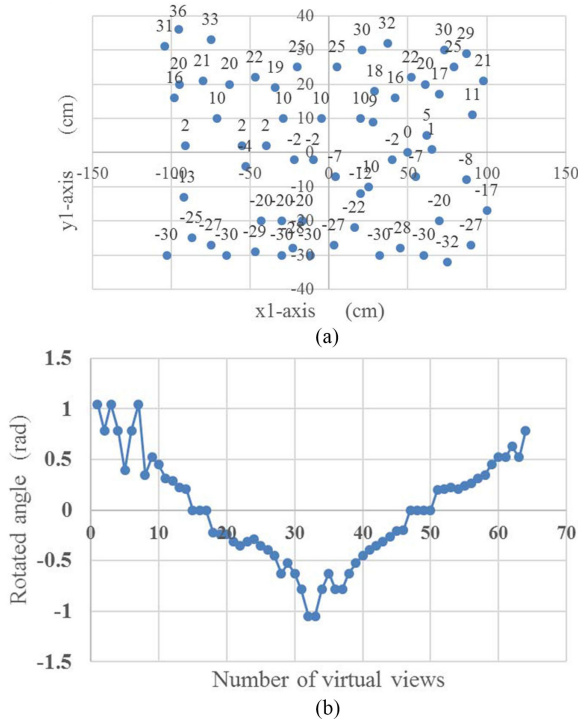


Fig. 10. The GCVVs for the experiments. (a) The position values (x_1, y_1) of the 64 cameras; (b) the camera rotation angles (θ_v) .

PSNR improvement is very significant after the third iteration, but the improvement becomes less significant with additional iterations. Another phenomenon is that the PSNR reaches the maximum in the third iteration, but the PSNR decreases to a stable value with more iterations. The same phenomena are exhibited by the MSSIM in Fig. 11(c). The PSNR and MSSIM decrement decrements are small because multiple virtual views are rendered and multiple cameras are optimized. Hence, the PSNR and MSSIM fluctuate during the first three iterations. The fluctuations are related to both the number of virtual views and cameras and the GCVV.

The PSNRs and MSSIMs are different for the different scenes. For example, the PSNR of the church scene is greater than that of the cat scene (approximately 2.0 dB) in Fig. 11(a) because of the differences in the complexity (i.e., geometrical and textual information) of the two scenes, as shown in Fig. 9(a) and (b).

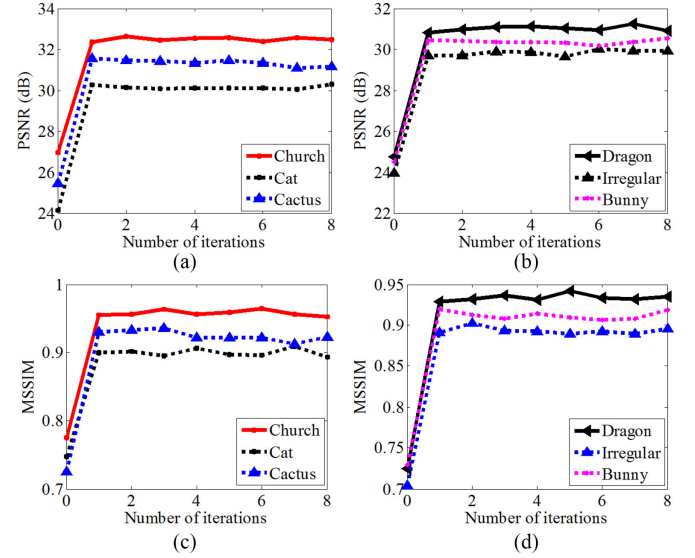


Fig. 11. The PSNRs and MSSIMs of the rendered views with respect to the number of optimization iterations for the six scenes. (a) and (c) The results for the church, cat and cactus scenes; (b) and (d) the results for the irregular, dragon and bunny scenes.

The incremental PSNR from the initial iteration to the eighth iteration is 6.3 dB for the church scene, 6.1 dB for the cat scene, and 5.8 dB for the cactus scene. The incremental MSSIM is 0.18 for the church scene, 0.16 for the cat scene, and 0.15 for the cactus scene. Therefore, the scene complexity also influences the optimal GCC.

Additionally, the experimental results shown in Fig. 11(b) and (d) yield the same conclusions as those drawn from Fig. 11(a) and (c). The PSNR and MSSIM improve dramatically after the third iteration but then improve more gradually with subsequent iterations. However, the PSNR and MSSIM fluctuations in these scenes are larger than those of the previous three scenes. The PSNRs and MSSIMs also differ among the three scenes. The rendering quality of the dragon scene is greater than the rendering qualities of the bunny and irregular scenes. For example, the PSNR of the dragon scene is greater than that of the bunny scene (0.5 dB) and irregular scene (1.5 dB). Additionally, the incremental PSNR is 6.7 dB for the dragon scene, 6.1 dB for the bunny scene, and the 5.8 dB for the irregular scene. The incremental MSSIM is 0.21 for the dragon

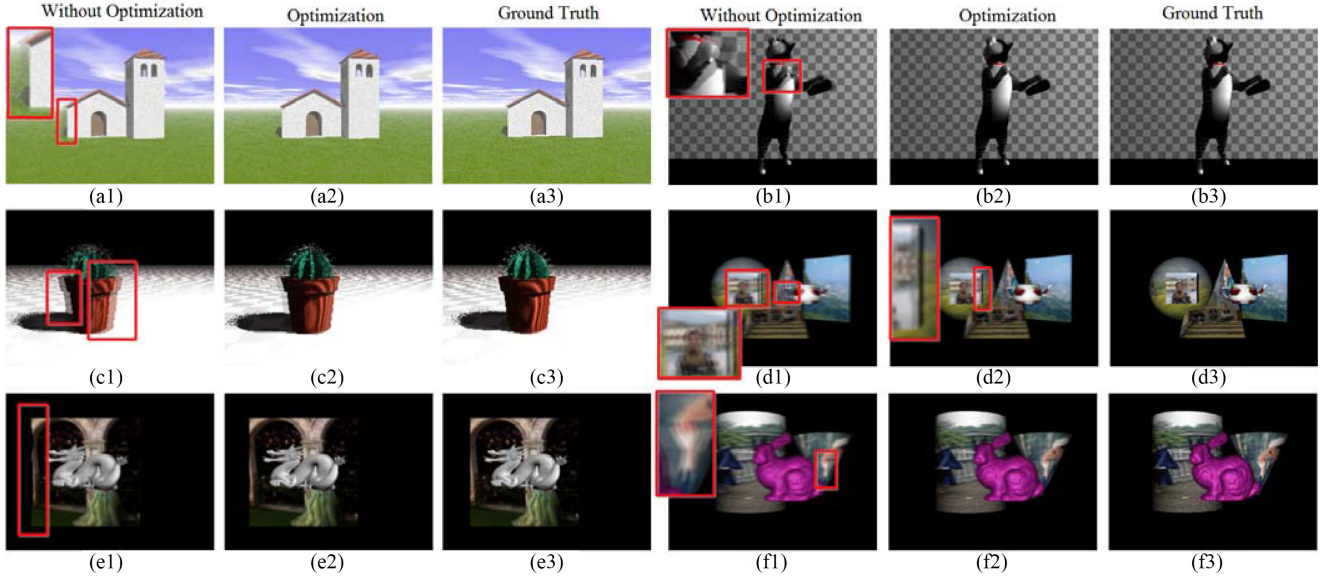


Fig. 12. Rendered images of the virtual views of the six scenes. The position of the virtual view is -2.0 cm. Diagrams (a1), (b1), (c1), (d1), (e1), (f1) illustrate the rendered virtual views without optimization; diagrams (a2), (b2), (c2), (d2), (e2), (f2) illustrate the rendered virtual views after eight iterations of the MFCM algorithm; (a3), (b3), (c3), (d3), (e3), (f3) show the ground truths.

scene, 0.19 for the bunny scene, and 0.18 for the irregular scene. Higher rendering quality is obtained because of decreased scene complexity.

The third phenomenon is that the rendered virtual views of the initial, i.e., Fig. 12(a1), iterations have some aliasing. This phenomenon is also observed in Fig. 12(b1), (c1), (d1), (e1), (f1) for the other scenes. The red boxes in the figures denote the areas of the aliasing. This phenomenon is due to an insufficient number of cameras being used. Here, the virtual views are rendered using only 32 cameras, thus implying a degree of under-sampling. If a larger number of cameras were used, the aliasing would not exist; then, the optimization performance of the MFCM algorithm could not be clearly compared. By comparing the initial and optimized rendering results, the optimization performance can be presented.

Moreover, the interpolation error of the rendered light ray is related to the difference between the GCVV and GCC. As shown in Table II and Fig. 10, differences remain between the GCVVs and the optimal GCCs. However, the rendering quality of the eighth iteration is better than the rendering quality of the initial results, which are not optimized. This phenomenon can also be observed in Fig. 12, where Fig. 12(a2) shows better results than Fig. 12(a1), Fig. 12(b2) shows better results than Fig. 12(b1), Fig. 12(c2) shows better results than Fig. 12(c1), Fig. 12(d2) shows better results than Fig. 12(d1), Fig. 12(e2) shows better results than Fig. 12(e1), and Fig. 12(f2) shows better results than Fig. 12(f1).

Finally, based on the above analysis, the rendering quality can be improved by simultaneously optimizing the position and direction of the GCCs with the MFCM algorithm.

B. Optimization Results for More Cameras and Virtual Views

In this experiment, we present the optimization results for the case of a large number of cameras and virtual views. The

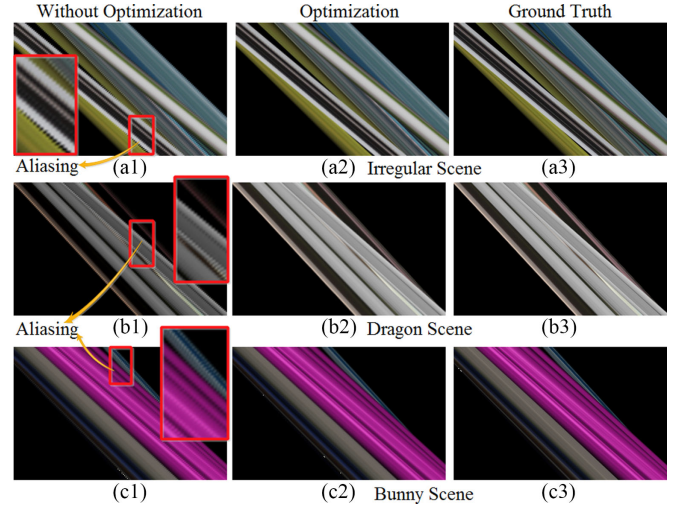


Fig. 13. The reconstructions of the synthetic EPI volumes for the irregular, dragon and bunny scenes. The reconstruction EPI volumes of the (a1)–(a3) irregular scene, (b1)–(b3) for the dragon scene, and (c1)–(c3) for the bunny scene. Diagrams (a1), (b1), (c1) show the reconstruction EPI volumes without optimization; diagrams (a2), (b2), (c2) illustrate the reconstruction EPI volumes after the ninth iteration of the MFCM algorithm.

irregular, dragon and bunny scenes shown in Fig. 9(d)–(f) are used to perform this experiment. Without loss of generality, let 200 virtual views be rendered by 100 cameras in a line. The virtual views are uniformly placed in the range $x \in [-100, 100]$ cm in a line and point in the same direction. Subsequently, these cameras are optimized using the MFCM algorithm. We find that the algorithm converges after nine iterations. The convergence time for the code running is about 0.130067 seconds.

We only present the rendered EPI volumes to describe the optimization results due to the large number of cameras and virtual views. From Fig. 13(a1), (b1), (c1), it can be observed that some aliasing in the reconstruction EPI volumes of the

TABLE III
THE GCVVs FOR THE COMPARISON EXPERIMENTS

| m | 1, 2, 3, 4, 5, 6, 7, 8, |
|------------------------|--|
| (x_1, y_1, θ_v) | $(-100, 0, \frac{\pi}{4}), (-93.45, 0, \frac{\pi}{5}), (-89.75, 0, \frac{\pi}{6}), (-83.67, 0, \frac{\pi}{7}), (-79.59, 0, \frac{\pi}{8}), (-77.11, 0, \frac{\pi}{9}), (-75.42, 0, \frac{\pi}{10}), (-70.34, 0, 0),$ |
| m | 9, 10, 11, 12, 13, 14, 15, 16, |
| (x_1, y_1, θ_v) | $(-65.18, 0, \frac{\pi}{11}), (-55.20, 0, \frac{\pi}{12}), (-52.04, 0, \frac{\pi}{13}), (-42.71, 0, 0), (-39.55, 0, \frac{\pi}{14}), (-33.7, 0, \frac{\pi}{15}), (-26.53, 0, 0), (-24.9, 0, 0),$ |
| m | 17, 18, 19, 20, 21, 22, 23, 24, 25, |
| (x_1, y_1, θ_v) | $(-17.28, 0, \frac{\pi}{16}), (-10.20, 0, \frac{\pi}{17}), (-8.12, 0, 0), (-2.04, 0, 0), (1.51, 0, 0), (6.12, 0, 0), (11.2, 0, 0), (13.57, 0, 0), (16.73, 0, 0),$ |
| m | 26, 27, 28, 29, 30, 31, 32, |
| (x_1, y_1, θ_v) | $(26.53, 0, -\frac{\pi}{17}), (30.61, 0, -\frac{\pi}{16}), (35.35, 0, -\frac{\pi}{15}), (42, 0, -\frac{\pi}{14}), (46.93, 0, -\frac{\pi}{13}), (56.57, 0, -\frac{\pi}{12}), (60.17, 0, -\frac{\pi}{11}),$ |
| m | 33, 34, 35, 36, 37, 38, 39, 40 |
| (x_1, y_1, θ_v) | $(63.23, 0, -\frac{\pi}{10}), (71.46, 0, -\frac{\pi}{9}), (77.51, 0, -\frac{\pi}{8}), (79.58, 0, -\frac{\pi}{7}), (87.75, 0, -\frac{\pi}{6}), (92.87, 0, -\frac{\pi}{5}), (95.84, 0, -\frac{\pi}{4}), (100, 0, -\frac{\pi}{4}).$ |

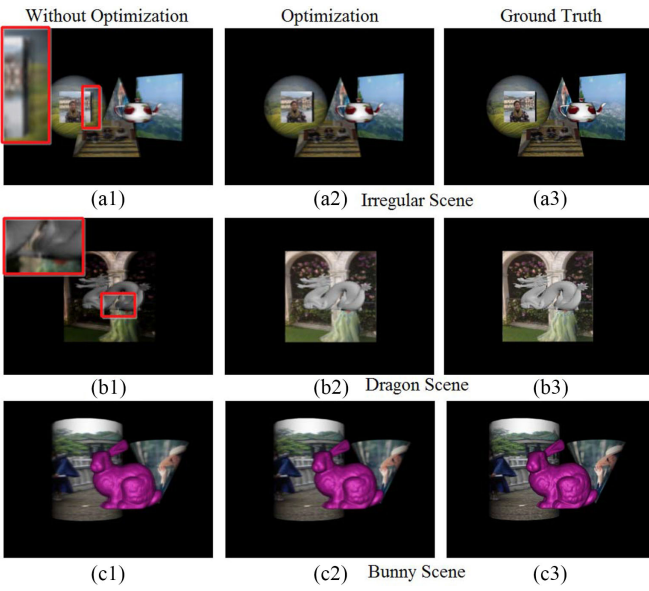


Fig. 14. Virtual views to be rendered for the irregular, dragon and bunny scenes. (a1), (b1), (c1) One of the 200 rendered virtual views without optimization. (a2), (b2), (c2) The corresponding rendered virtual views after nine iterations of the MFCM algorithm. The red boxes indicate aliasing. (a3), (b3), (c3) The ground truths. The rendered virtual views of the (a1)–(a3) irregular, (b1)–(b3) dragon, and (c1)–(c3) bunny scenes.

three scenes exists when the pre-optimization GCC is used. The aliasing is reduced after the ninth iteration of the MFCM algorithm, as shown in Fig. 13(a2), (b2), (c2). This phenomenon can also be observed from one of the 200 rendered virtual views shown in Fig. 14. The red boxes in the figures denote the areas of aliasing. Apparently, the aliasing in Fig. 14(a1), (b1), (c1) is more serious than in Fig. 14(a2), (b2), (c2). Furthermore, the PSNRs in Fig. 14(a1) and (a2) are 27.76 dB and 34.63 dB, respectively. The MSSIMs in Fig. 14(a1) and (a2) are 0.8432 and 0.9507, respectively. The PSNRs in Fig. 14(b1) and (b2) are 27.29 dB and 33.82 dB, respectively. The MSSIMs in Fig. 14(b1) and (b2) are 0.8131 and 0.9390, respectively. The PSNRs in Fig. 14(c1) and (c2) are 29.56 dB and 35.04 dB, respectively. The MSSIMs in Fig. 14(c1) and (c2) are 0.8709 and 0.9783, respectively. Therefore, the MFCM algorithm also optimizes the GCC for a large number of cameras and virtual views.

C. Comparison of Results From MFCM and ARC Algorithms

The ARC algorithm [16] is selected for comparison purposes. The ARC algorithm is applied to optimize a set of camera positions on a plane. This algorithm assumes that the cameras point in the same direction. The irregular, dragon and bunny scenes in Fig. 9(d)–(f) are used to compare the proposed MFCM algorithm and the ARC algorithm. Seventeen cameras are used to capture the three scenes. Forty virtual views are rendered using the captured images. The virtual views are randomly placed, as shown in Table III. The GCCs are uniformly initialized, and then the GCCs of the seventeen cameras are optimized using both the MFCM and ARC algorithms.

Table IV presents the optimization results for the seventeen cameras. The optimization results based on the MFCM algorithm and ARC algorithm are different. For example, the optimal position for the first camera is -98.42 cm when the MFCM algorithm is used, whereas the optimal position is -97.6 cm when the ARC algorithm is used for the irregular scene. The optimal positions for the other cameras also differ because of the differences between the two algorithms. However, the differences between the optimal results are relatively small. Thus, the proposed optimization algorithm has the same optimization effectiveness as the ARC algorithm for the positions of the cameras. Unfortunately, the ARC algorithm is not able to optimize the directions of the cameras (i.e., the directions of cameras in Table IV are zero). In contrast, the MFCM algorithm can optimize the directions and positions of the GCC simultaneously.

The PSNRs and MSSIMs of the experimental comparison are shown in Fig. 15. The figure shows that for both the ARC and MFCM algorithms, the PSNRs and MSSIMs improve significantly after the first iteration, although the improvement tends to decrease as the iterations continue. The optimal rendering quality is similar for both algorithms. The PSNRs of the ARC and the MFCM algorithms are 28.1 and 30.3 dB, respectively, in Fig. 15(a). The MSSIMs of the same scene are 0.8 and 0.9 in Fig. 15(e). Additionally, the incremental PSNR from the ARC is approximately 3.1 dB, and the incremental MSSIM is 0.34. The incremental PSNR from the MFCM algorithm is approximately 5.3 dB, and the incremental MSSIM is approximately 0.24. A similar phenomenon can be observed in Fig. 15(b), (c), (f), (g) for the dragon and bunny scenes. Therefore, the MFCM

TABLE IV
THE OPTIMAL VALUES OF THE GCCs USING THE METHODS OF ARC AND MFCM

| Methods | n | 1, | 2, | 3, | 4, | 5, | 6, | 7, | 8, | |
|---------|-----------|--|-----|-----|-----|-----|-----|-----|-----|----|
| ARC | Irregular | $(-98.42, 0, 0), (-78.73, 0, 0), (-60.56, 0, 0), (-51.26, 0, 0), (-34.4, 0, 0), (-21.1, 0, 0), (-13.85, 0, 0), (-2.33, 0, 0),$ | | | | | | | | |
| | Dragon | $(-99.05, 0, 0), (-79.54, 0, 0), (-60.56, 0, 0), (-50.9, 0, 0), (-34.7, 0, 0), (-21.3, 0, 0), (-12.45, 0, 0), (-1.96, 0, 0),$ | | | | | | | | |
| | Bunny | $(-98.42, 0, 0), (-78.73, 0, 0), (-62.09, 0, 0), (-52.36, 0, 0), (-35.94, 0, 0), (-20.29, 0, 0), (-13.85, 0, 0), (-2.58, 0, 0),$ | | | | | | | | |
| | n | 9, | 10, | 11, | 12, | 13, | 14, | 15, | 16, | 17 |
| | Irregular | $(5.79, 0, 0), (12.19, 0, 0), (24.57, 0, 0), (32.85, 0, 0), (43.92, 0, 0), (52.81, 0, 0), (66.38, 0, 0), (86.9, 0, 0), (91.46, 0, 0)$ | | | | | | | | |
| | Dragon | $(6.57, 0, 0), (12.19, 0, 0), (24.57, 0, 0), (33.92, 0, 0), (42.22, 0, 0), (53.74, 0, 0), (66.38, 0, 0), (86.9, 0, 0), (93.0, 0, 0)$ | | | | | | | | |
| | Bunny | $(4.9, 0, 0), (11.04, 0, 0), (25.78, 0, 0), (32.85, 0, 0), (43.92, 0, 0), (52.81, 0, 0), (67.78, 0, 0), (87.65, 0, 0), (92.02, 0, 0)$ | | | | | | | | |
| MFCM | n | 1, | 2, | 3, | 4, | 5, | 6, | 7, | 8, | 9, |
| | | $\left(-97.6, 0, \frac{\pi}{4}\right), \left(-79.43, 0, \frac{\pi}{6}\right), \left(-60.56, 0, \frac{\pi}{8}\right), \left(-52.4, 0, \frac{\pi}{10}\right), \left(-33.3, 0, \frac{\pi}{11}\right), \left(-22.9, 0, \frac{\pi}{13}\right), \left(-12.65, 0, \frac{\pi}{16}\right),$ | | | | | | | | |
| | | $\left(-3.3, 0, 0\right), \left(5.75, 0, 0\right),$ | | | | | | | | |
| | n | 10, | 11, | 12, | 13, | 14, | 15, | 16, | 17 | |
| | | $\left(12.78, 0, 0\right), \left(24.9, 0, -\frac{\pi}{15}\right), \left(32.21, 0, -\frac{\pi}{13}\right), \left(43.67, 0, -\frac{\pi}{12}\right), \left(52.79, 0, -\frac{\pi}{11}\right), \left(66.38, 0, -\frac{\pi}{8}\right), \left(86.48, 0, -\frac{\pi}{7}\right),$ | | | | | | | | |
| | | $\left(92.09, 0, -\frac{\pi}{4}\right)$ | | | | | | | | |

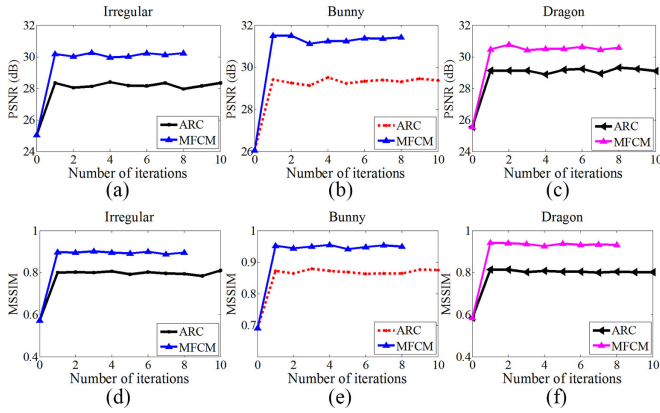


Fig. 15. Diagrams that show the PSNRs and MSSIMs of the rendering results based on the ARC and MFCM algorithms. (a) and (e) The results for the irregular scene. (b) and (f) The results for the dragon scene. (c) and (g) The results for the bunny scene.



Fig. 16. (a) Plaster model scene. (b) Miniascape scene. (c) Data acquisition setup. A hand-held camera can be moved along a track plane.

algorithm offers better optimization effectiveness than the ARC algorithm when the cameras and virtual views are rotated. The reason for this behavior can be understood by considering the fact that the ARC algorithm can only optimize the camera positions. Furthermore, the ARC algorithm converges after ten iterations, whereas the MFCM algorithm converges after eight iterations. The convergence time is approximately 0.041014 seconds for the ARC algorithm and approximately 0.031038 seconds for the MFCM algorithm. Therefore, the convergence speed of the MFCM algorithm is faster than that of the ARC algorithm. The reason for this is that each rendered view is split into a set of rendered light rays for the ARC algorithm.

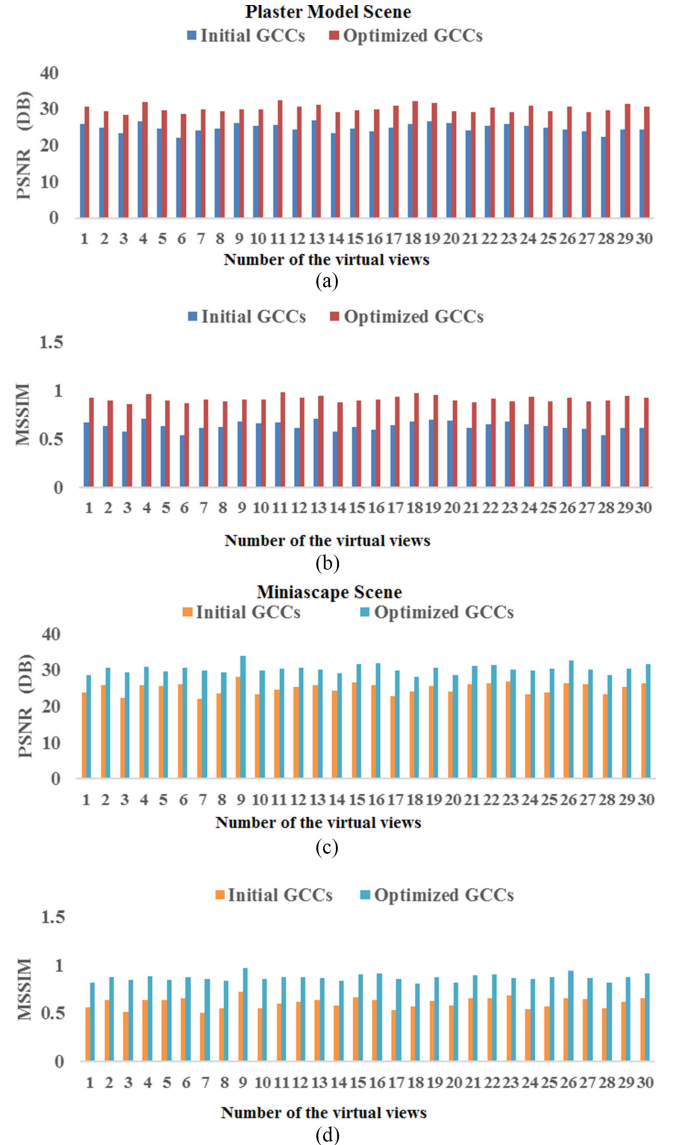


Fig. 17. The PSNRs and MSSIMs of the rendered virtual views using the initial GCC and optimized GCC obtained using the MFCM algorithm. (a) and (b) The results for the plaster model scene; (c) and (d) the results for the miniascape scene.

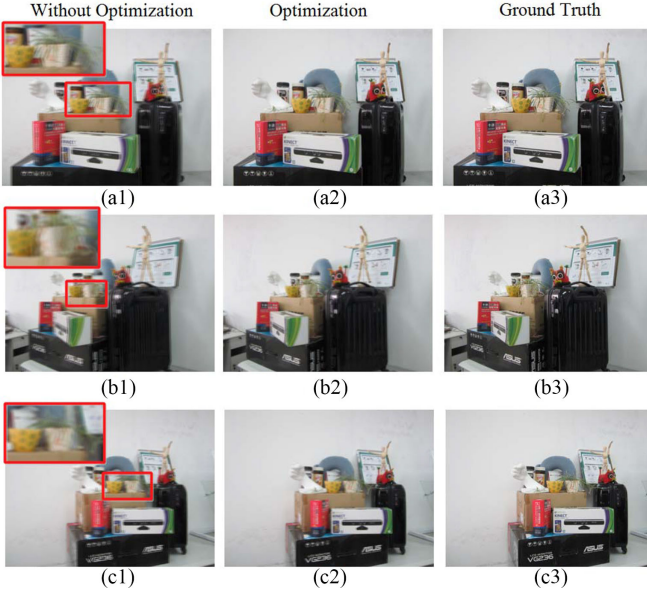


Fig. 18. (a1), (b1), (c1) Three rendered multi-views of the plaster-model scene without optimization for significant aliasing exists. (a2), (b2), (c2) The corresponding rendered virtual views after optimizing the GCCs using the MFCM algorithm. (a3), (b3), (c3) The ground truths.

However, the rendering error of each rendered view is quantified based on the NCF area for the MFCM algorithm. Calculating the NCF area is easier than calculating the rendering errors of all the rendered light rays.

D. Actual Scenes

To evaluate the MFCM algorithm, the optimization experiments are also performed using the two actual scenes depicted in Fig. 16(a) and (b). For each scene, twenty GCCs are initialized and optimized in advance, and then we use a hand-held camera (see Fig. 16(c)) with these GCCs to capture multi-view images. Thirty virtual views are rendered using the captured images. The GCVVs are randomly placed, i.e., the positions of the virtual views are random, and the virtual views do not point in the same direction.

The experimental results for the actual scenes yield the same conclusions as those for the synthetic scenes. As depicted in Fig. 17, the rendering qualities of the rendered views obtained with the optimized GCCs are greater than the rendering qualities with the initial GCCs. This phenomenon is also observed from the rendered multi-views shown in Fig. 18(a1), in which the reconstructed view exhibits some obvious ghosting and aliasing. The ghosting and aliasing occur because the scene complexity, such as the leafage and irregular shape, is challenging to reconstruct accurately when the GCCs are uniformity initialized. However, when the GCCs are optimized, the ghosting and aliasing in the rendered views decreases, as shown in Fig. 18(a2). The same phenomenon can be observed when comparing Fig. 18(b1), (c1) with Fig. 18(b2), (c2) and Fig. 19(a1), (b1), (c1) with Fig. 19(a2), (b2), (c2). These results also suggest that the MFCM algorithm can be applied to optimize the GCCs in the actual scenes.

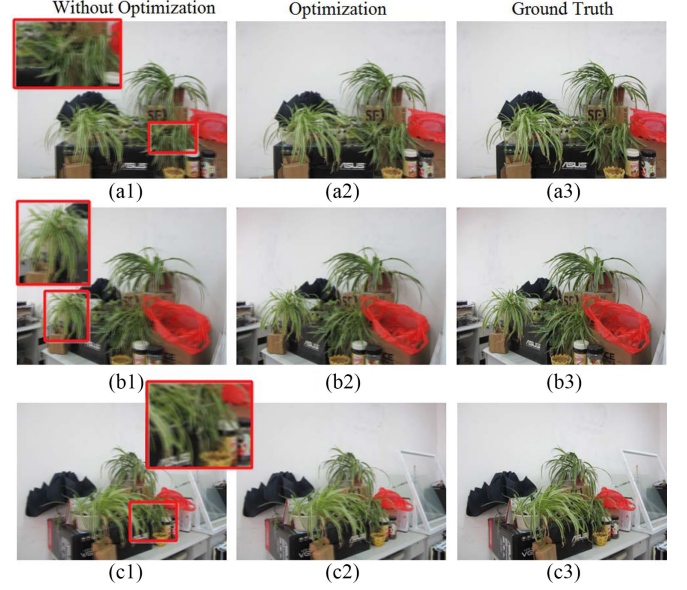


Fig. 19. (a1), (b1), (c1) Three rendered multi-views of the miniascape scene without optimization for significant aliasing exists. (a2), (b2), (c2) The corresponding rendered virtual views after optimizing the GCCs. (a3), (b3), (c3) The ground truths.

VI. CONCLUSION

This study proposes a NCF model and uses it to describe the topology of the intersecting coverage range of the fields of view between a camera and a virtual view. Then, we derive a mathematical model to quantify the relationship between the rendering quality and the GCC. The NCF model is applied to assess the quality of the virtual view. We demonstrate that the NCF model can greatly simplify the calculation complexity of the GCC optimization. Furthermore, a new MFCM optimization algorithm is proposed. Using the MFCM algorithm and NCF model, the positions and directions of the cameras are simultaneously optimized. The rendering quality is improved by optimizing the GCCs in the IBR system. It is worth noting that this algorithm depends on a minimal amount of geometric information.

APPENDIX A PROOF OF THEOREM 1

Starting from the relationship between Λ_{nm} and the rendering error, the influence of the GCC variation on Λ_{nm} is analyzed. For each virtual view, there are N NCFs; however, we select only the minimum Λ_{nm} to serve as the NCF of the virtual view. Let

$$\tilde{\Lambda}_m = \min_{n=1, \dots, N} \Lambda_{nm} \quad (23)$$

be the minimum Λ_{nm} between the virtual view V_m and all the capturing cameras. Intuitively, we find in Fig. 1 that when the GCC is changed, the variation of $\tilde{\Lambda}_m$ is the same as the rendering error of the virtual view. That is, as $\tilde{\Lambda}_m$ decreases, the rendering error decreases. Additionally, compared with the rendering error, $\tilde{\Lambda}_m$ is simpler and more effective for describing the GCC variation. However, the relationship between $\tilde{\Lambda}_m$ and the rendering error, is very complex, which makes the calculation of

$\tilde{\Lambda}_m$ difficult. Fortunately, the above analysis reveals that there is indeed a linear approximation between $\tilde{\Lambda}_m$ and the rendering error. Based on an approximation, we can assume the following

$$e_m = \Phi_m(\tilde{\Lambda}_m) \approx \kappa_m \tilde{\Lambda}_m. \quad (24)$$

Therefore, we apply $\tilde{\Lambda}_m$ instead of the rendering error to optimize the GCC. Substituting (24) to (1), the optimal geometric configuration of N cameras can be rewritten as a function of $\tilde{\Lambda}_m$ as

$$\hat{\mathbf{C}}(\hat{x}, \hat{y}, \hat{z}, \hat{\theta}_x, \hat{\theta}_y, \hat{\theta}_z) = \arg \min_{\mathbf{C}(x, y, z, \theta_x, \theta_y, \theta_z)} \sum_{m=1}^M \kappa_m \tilde{\Lambda}_m. \quad (25)$$

APPENDIX B PROOF OF THEOREM 2

Firstly, we derive the rendering error of the rendered light ray with the GCC and GCVV using (7) and (8). As shown in Fig. 5, obviously, the rendering error Δv_j of l_j is related to u_i and v_j . In Fig. 5, $p(x, y)$ is the point that the desired light ray l_j intersects the object surface. By (7), u_i is calculated using the following expression [44]

$$u_i = f \left(\frac{(x - x_0) \cos \theta_c + (y - y_0) \sin \theta_c}{(x_0 - x) \sin \theta_c + (y - y_0) \cos \theta_c} \right). \quad (26)$$

Similarly, by (8), the expression of v_j is obtained as

$$v_j = f \left(\frac{(x - x_1) \cos \theta_v + (y - y_1) \sin \theta_v}{(x_1 - x) \sin \theta_v + (y - y_1) \cos \theta_v} \right). \quad (27)$$

By (26) and (27), the rendering error of the desired light ray l_j with the light ray l_i can be expressed as

$$\Delta v_j = |u_i - v_j|. \quad (28)$$

In (28), as the distance between the points u_i and v_j decreases, Δv_j also decreases. Additionally, we have previously defined the field-of-view of the virtual view as consisting of L rendered light rays. Then, the rendering error e_m of the virtual view $V_m(x_1, y_1, \theta_v)$ with the camera $C_n(x_0, y_0, \theta_c)$ is calculated by considering all of the rendered light rays. Here, we can use the MSE of the desired light rays to calculate the rendering error of the virtual view. As shown in Fig. 4, the left edge of camera intersects the straight line $y = Z_c$ at the point $A_1(a_1, Z_c)$. $A_L(a_L, Z_c)$ is the intersection point for the right edge. Here, a_1 and a_L can be calculated by (7) as

$$\begin{cases} a_1 = x_0 + \frac{(y_0 - Z_c)(u_0 \cos \theta_c + f \sin \theta_c)}{f \cos \theta_c - u_0 \sin \theta_c}, \\ a_L = x_0 + \frac{(Z_c - y_0)(u_0 \cos \theta_c - f \sin \theta_c)}{f \cos \theta_c + u_0 \sin \theta_c}. \end{cases} \quad (29)$$

Similarly, the left edge of virtual view intersects the straight line $y = Z_c$ at the point $B_1(b_1, Z_c)$. The point $B_L(b_L, Z_c)$ is the intersection for the right edge. Here, b_1 and b_L can be calculated by (8) as

$$\begin{cases} b_1 = x_1 + \frac{(y_1 - Z_c)(v_0 \cos \theta_v + f \sin \theta_v)}{f \cos \theta_v - v_0 \sin \theta_v}, \\ b_L = x_1 + \frac{(Z_c - y_1)(v_0 \cos \theta_v - f \sin \theta_v)}{f \cos \theta_v + v_0 \sin \theta_v}. \end{cases} \quad (30)$$

Additionally, the intersection (x_j, Z_c) that the desired light ray l_j intersects to the straight line $y = Z_c$ is calculated as

$$x_j = x_1 + \frac{(Z_c - y_1)(v_j \cos \theta_v - f \sin \theta_v)}{(f \cos \theta_v + v_j \sin \theta_v)}. \quad (31)$$

If $x_j < a_1$ or $x_j > a_L$, that the camera $C_n(x_0, y_0, \theta_c)$ has not light ray to cover the j th desired light ray of $V_m(x_1, y_1, \theta_v)$. In this case, the rendering error considered to be a constant P which depend on the size of the j th desired light ray. Therefore, the rendering error of the j th desired light ray l_j of the $V_m(x_1, y_1, \theta_v)$ with light ray l_i can be rewritten as

$$\Delta v_j = \begin{cases} |u_i - v_j|, & x_j \geq a_1 \text{ or } x_j \leq a_L \\ P, & x_j < a_1 \text{ or } x_j > a_L, \end{cases} \quad \Delta v_j \ll P. \quad (32)$$

According to the definition of MSE, the rendering error of $V_m(x_1, y_1, \theta_v)$ with $C_n(x_0, y_0, \theta_c)$ is expressed as

$$e_m = \Phi_\Delta(x_0, y_0, \theta_c) = \sqrt{\frac{1}{L} \cdot \sum_{j=1}^L (\Delta v_j)^2}. \quad (33)$$

In (33), the value of e_m also depends on the $x_0, x_1, y_0, y_1, \theta_c$, and θ_v . As the difference between (x_1, y_1, θ_v) and (x_0, y_0, θ_c) decreases, the value of e_m also decreases.

We next derive the NCF area function which is calculated by the GCC and GCVV. As shown in Fig. 4, the constant depth line $y = Z_c$ limits the NCF area; therefore, the area of the CF is calculated by the intersection points, A_1, A_L, B_1 and B_L . Their coordinate expressions are presented by (29) and (30). Then, the NCF area is expressed as

$$\begin{aligned} S_{nm} &= \Phi_s(x_0, y_0, \theta_c, x_1, y_1, \theta_v) \\ &= S_v + S_c - 2 \begin{cases} S_I, & x_1 < x_0 < x_1 + W, \\ S_{II}, & x_1 - W < x_0 \leq x_1, \end{cases} \end{aligned} \quad (34)$$

where $x_0 \in [x_1 - W, x_1 + W]$, $W = 2Z_c u_0 / f$, S_I represents the CF area in the case for the camera on the right side of virtual view, and S_{II} is for the camera on the left side of virtual view. Denote S_I as

$$S_I = \begin{cases} (S_1 - S_2) \text{ or } S_v, & \text{if } a_1 < b_1, a_L \geq b_L \\ (S_1 - S_3) \text{ or } S_c, & \text{if } a_1 \geq b_1, a_L < b_L \\ (S_1 - S_2 - S_3) \text{ or } (S_v - S_3), & \text{if } a_1 < b_1, b_1 < a_L < b_L \\ S_1, & \text{if } b_1 \leq a_1 < b_L, a_L \geq b_L \\ 0, & \text{otherwise} \end{cases} \quad (35)$$

where S_1 is the area of triangle $\Delta A_1 Q_2 B_L$, S_2 is the area of triangle $\Delta A_1 Q_1 B_1$, and S_3 is the area of triangle $\Delta A_L Q_3 B_L$

in Fig. 4. Similarly, denote S_{II} as

$$S_{II} = \begin{cases} (S_4 - S_5) \text{ or } S_c, & \text{if } a_1 > b_1, a_L \leq b_L \\ (S_4 - S_6) \text{ or } S_v, & \text{if } a_1 \leq b_1, a_L > b_L \\ (S_4 - S_5 - S_6) \text{ or } (S_v - S_5), & \text{if } b_1 < a_1 < b_L, a_L > b_L \\ S_4, & \text{if } a_1 \leq b_1, b_1 < a_L \leq b_L \\ 0, & \text{otherwise} \end{cases} \quad (36)$$

where S_4 is the area of triangle $\Delta B_1 Q_2 A_L$, S_5 is the area of triangle $\Delta B_1 Q_1 A_1$, and S_6 is the area of triangle $\Delta B_L Q_3 A_L$. In (34), S_{nm} is determined by $x_0, x_1, y_0, y_1, \theta_c$, and θ_v . As the difference between (x_1, y_1, θ_v) and (x_0, y_0, θ_c) decreases, S_{nm} decreases.

Based on (33) and (34), we prove the quantification of NCF area to quantify the rendering error. It can find in (34) that the camera position x_0 can be represented using S_{nm} from the NCF area function $S_{nm} = \Phi_s(x_0, y_0, \theta_c, x_1, y_1, \theta_v)$. It can be written as

$$x_0 = \Phi_t(S_{nm}) = \begin{cases} t_I, x_1 < x_0 < x_1 + W \\ t_{II}, x_1 - W < x_0 \leq x_1 \end{cases}, \quad (37)$$

where t_I and t_{II} are the camera positions which are calculated by the NCF area. Their derivations mainly depend on the function $S_{nm} = \Phi_s(x_0, y_0, \theta_c, x_1, y_1, \theta_v)$ by (34). So, the expressions of t_I and t_{II} can be respectively represented as

$$t_I = \begin{cases} t_1, a_1 < b_1, a_L \geq b_L \\ t_2, a_1 \geq b_1, a_L < b_L \\ t_3, a_1 < b_1, b_1 < a_L < b_L \\ t_4, b_1 \leq a_1 < b_L, a_L \geq b_L \\ \Psi, \text{ otherwise} \end{cases}, \quad (38)$$

$$t_{II} = \begin{cases} t_5, a_1 > b_1, a_L \leq b_L \\ t_6, a_1 \leq b_1, a_L > b_L \\ t_7, b_L > a_1 > b_1, a_L > b_L \\ t_8, a_1 \leq b_1, b_1 < a_L \leq b_L \\ \Psi, \text{ otherwise} \end{cases}, \quad (39)$$

where

$$\begin{cases} t_1 = \Phi_s^{-1}(S_c + S_v - 2(S_1 - S_2) \text{ or } S_v) \\ t_2 = \Phi_s^{-1}(S_c + S_v - 2(S_1 - S_3) \text{ or } S_c) \\ t_3 = \Phi_s^{-1}(S_c + S_v - 2(S_1 - S_2 - S_3) \text{ or } (S_v - S_3)) \\ t_4 = \Phi_s^{-1}(S_c + S_v - 2S_1) \\ t_5 = \Phi_s^{-1}(S_c + S_v - 2(S_4 - S_5) \text{ or } S_c) \\ t_6 = \Phi_s^{-1}(S_c + S_v - 2(S_4 - S_6) \text{ or } S_v) \\ t_7 = \Phi_s^{-1}(S_c + S_v - 2(S_4 - S_5 - S_6) \text{ or } (S_v - S_5)) \\ t_8 = \Phi_s^{-1}(S_c + S_v - 2S_4) \end{cases}, \quad (40)$$

where $\Phi_s^{-1}(\cdot)$ is the inverse function of $\Phi_s(\cdot)$. Furthermore, Ψ represents that x_0 is not calculated by NCF area (i.e., the CF area is zero). Finally, substituting (37) into (33), we can obtain the following expression of the rendering error function

$$e_m = \Phi_{\Delta v}(S_{nm}) = \Phi_{\Delta}(\Phi_t(S_{nm})) \approx \omega_m \min_{n=1, \dots, N} S_{nm}. \quad (41)$$

The above function (i.e., $\Phi_{\Delta v}(\cdot)$) is applied to describe the relationship between the rendering error and the NCF area. It can be found that the variation of the rendering error is also described by the NCF area. Additionally, by (33) and (34), we consider that when the GCC is changed, as the NCF area decreases, the rendering error decreases. Therefore, the rendering error can be quantified by the NCF area.

APPENDIX C

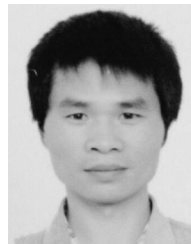
THE EXPERIMENTS RESULTS FOR THE QUANTIFICATION

Please see supplemental material at <http://ieeexplore.ieee.org> or http://ei.hust.edu.cn/lab/mcnc-lab/index.html#/resource/other/TMM_NCF_paper_supplemental_material.pdf.

REFERENCES

- [1] H.-Y. Shum, S. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.
- [2] C. Zhang and T. Chen, "A survey on image-based rendering C representation, sampling and compression," *EURASIP Signal Process., Image Commun.*, vol. 19, no. 1, pp. 1–28, Jan. 2004.
- [3] M. N. Do, Q. H. Nguyen, H. T. Nguyen, D. Kubacki, and S. J. Patel, "Immersive visual communication," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 58–66, Jan. 2011.
- [4] A. Kubota *et al.*, "Multi-view imaging and 3DTV," *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 10–21, Nov. 2007.
- [5] T. Maugey and P. Frossard, "Interactive multiview video system with low complexity 2D look around at decoder," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1070–1082, Aug. 2013.
- [6] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 67–76, Jan. 2011.
- [7] L. HanShin and P. HyunWook, "An efficient multi-view generation method from a single-view video based on affine geometry information," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 726–737, Apr. 2014.
- [8] H.-Y. Shum, S. C. Chan, and S. B. Kang, *Image-Based Rendering*. Vienna, Austria: Springer, 2007, pp. 71–90.
- [9] N. King-To *et al.*, "A multi-camera approach to image-based rendering and 3D/multiview display of ancient Chinese artifacts," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1631–1641, Dec. 2012.
- [10] P. Hedman, T. Ritschel, G. Drettakis, and G. Brostow, "Scalable inside-out image-based rendering," *ACM Trans. Graph.*, vol. 35, no. 6, 2016, Art. no. 231.
- [11] J. Navarro and A. Buades, "Robust and dense depth estimation for light field images," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1873–1886, Apr. 2017.
- [12] D. Cho, S. Kim, Y. W. Tai, and I. S. Kweon, "Automatic trimap generation and consistent matting for light-field images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1504–1517, Aug. 2017, doi: 10.1109/TPAMI.2016.2606397.
- [13] R. Jain, J. Grzyb, and U. R. Pfeiffer, "Terahertz light-field imaging," *IEEE Trans. THz Sci. Technol.*, vol. 6, no. 5, pp. 649–657, Sep. 2016.
- [14] M. W. Tao *et al.*, "Shape estimation from shading, defocus, and correspondence using light-field angular coherence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 546–560, Mar. 2017.
- [15] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet Transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 133–147, 2018.
- [16] J. X. Chai, X. Tong, S. C. Chan, and H.-Y. Shum, "Plenoptic sampling," in *Proc. Annu. Conf. Comput. Graph.*, Jul. 2000, pp. 307–318.

- [17] C. Zhang and T. Chen, "Spectral analysis for sampling image-based rendering data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1038–1050, Nov. 2003.
- [18] M. N. Do, D. Marchand-Maillet, and M. Vetterli, "On the bandwidth of the plenoptic function," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 708–717, Feb. 2012.
- [19] C. Gilliam, P. Dragotti, and M. Brookes, "On the spectrum of the plenoptic function," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 502–516, Feb. 2014.
- [20] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 1996, pp. 31–42.
- [21] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 1996, pp. 43–54.
- [22] C. Zhang and T. Chen, "Active rearranged capturing of image-based rendering scenes-theory and practice," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 520–531, Apr. 2007.
- [23] L. Zucheul and Q. N. Truong, "Multi-array camera disparity enhancement," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2168–2177, Dec. 2014.
- [24] F. Safaei *et al.*, "Scene-adaptive configuration of two cameras using the correspondence field function," in *Proc. IEEE Int. Conf. Multimedia Expo.*, PA, USA, Jul. 2013, pp. 1–6.
- [25] J. Berent and P. L. Dragotti, "Plenoptic manifolds," *IEEE. Signal Process. Mag.*, vol. 24, no. 6, pp. 34–44, Nov. 2007.
- [26] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen, "Unstructured lumigraph rendering," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 2001, pp. 425–432.
- [27] H. T. Nguyen and M. N. Do, "Error analysis for image-based rendering with depth information," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 703–716, Apr. 2009.
- [28] S. X. Liu *et al.*, "On the relationship between multi-view data capturing and quality of rendered virtual view," *Imag. Sci. J.*, vol. 57, no. 5, pp. 250–259, 2009.
- [29] H. Schirmacher, W. Heidrich, and H. P. Seidel, "Adaptive acquisition of lumigraphs from synthetic scenes," *Comput. Graph. Forum*, vol. 18, no. 3, pp. 151–160, Sep. 1999.
- [30] H. Shidanshidi, F. Safaei, and W. Li, "A quantitative approach for comparison and evaluation of light field rendering techniques," in *Proc. IEEE Int. Conf. Multimedia Expo.*, PA, USA, Jul. 2011, pp. 1–4.
- [31] S. Fleishman, D. Cohen-Or, and D. Lischinski, "Automatic camera placement for image-based modeling," *Comput. Graph. Forum*, vol. 19, no. 2, Jun. 2000, pp. 101–110.
- [32] T. Werner, V. Hlav'ac, A. Leonardis, and T. Pajdla, "Selection of reference views for image-based representation," in *Proc. IEEE 13th Int. Conf. Pattern Recogn.*, Aug. 1996, pp. 73–77.
- [33] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proc. 27th Annu. Conf. Comput. Graphics Interactive Techniques*. New York, NY, USA: ACM Press/Addison-Wesley, Jul. 2000, pp. 297–306.
- [34] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, "Using plane+parallax for calibrating dense camera arrays," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, Jun. 2004, vol. 1, pp. 1–2.
- [35] K. Takahashi and T. Naemura, "Unstructured light field rendering using on-the-fly focus measurement," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2005, pp. 1–4.
- [36] K. Takahashi and T. Naemura, "Layered light-field rendering with focus measurement," *Signal Process., Image Commun.*, vol. 21, no. 6, pp. 519–530, 2006.
- [37] C. Chen and D. Schonfeld, "Geometrical plenoptic sampling," in *Proc. 16th IEEE Int. Conf. Image Process.*, 2009, pp. 3769–3772.
- [38] N. Snavely, R. Garg, M. S. Steven, and R. Szeliski, "Finding paths through the world's photos," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008, Art. no. 15.
- [39] A. Davis, M. Levoy, and F. Durand, "Unstructured light fields," *Comput. Graph. Forum*, vol. 31, no. 2, May 2012, pp. 305–314.
- [40] H. Shidanshidi, F. Safaei, and W. Li, "Objective evaluation of light field rendering methods using effective sampling density," in *Proc. IEEE 13th Int. Workshop Multimedia Signal Process.*, Hangzhou, China, Oct. 2011, pp. 1–6.
- [41] H. Shidanshidi, F. Safaei, and W. Li, "Estimation of signal distortion using effective sampling density for light field-based free viewpoint video," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1677–1693, Oct. 2015.
- [42] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.
- [43] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge U.K.: Cambridge Univ. Press, 2003.
- [44] R. Bolles, H. Baker, and D. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 7–55, 1987.
- [45] R. N. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-mean model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.
- [46] H. Choe and J. B. Jordan, "On the optimal choice of parameters in a fuzzy c-means algorithm," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 1992, pp. 349–354.
- [47] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 1, pp. 1–8, Jan. 1980.
- [48] W. I. Zangwill, "Non-linear programming via penalty functions," *Manage. Sci.*, vol. 13, no. 5, pp. 344–358, 1967.
- [49] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence theory for fuzzy c-means: Counterexamples and repairs," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-17, no. 5, pp. 873–877, Sep./Oct. 1987.
- [50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.



Changjian Zhu received the B.S. degree in electronics and information engineering and the M.S. degree in electronic circuits and systems from Guangxi Normal University, Guilin, China, in 2006 and 2009, respectively. He is currently working toward the Ph.D. degree in the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. His research interests include light field sampling, and applications of sampling theory to image processing, in particular, image-based rendering.



Li Yu (M'08) received the B.S., M.S., and Ph.D. degrees in electronic and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1992, 1995, and 1999, respectively. She is currently a Professor in the School of Electronic Information and Communications, Huazhong University of Science and Technology. She is the Director of the Research Center of Broadband Wireless Communication and Multimedia Communication. Her research interests include multimedia signal processing, multimedia communication, and wireless networks.



Zixiang Xiong (S'91–M'96–SM'02–F'07) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1996. He is currently a Professor in the Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA. His main research interests include image/video processing, networked multimedia, and multiuser information theory. Prof. Xiong received the NSF Career Award in 1999, the ARO Young Investigator Award in 2000, and the ONR Young Investigator Award in 2001. He is the co-recipient of the 2006 IEEE Signal Processing Magazine Best Paper Award, Top 10% Paper Awards at the 2011 and 2015 IEEE Multimedia Signal Processing Workshops, and the IBM Best Student Paper Award at the 2016 IEEE International Conference on Pattern Recognition. He was the Publications Chair of the International Conference on Acoustics, Speech, and Signal Processing 2007, the Technical Program Committee Co-Chair of the Information Theory Workshop 2007, the Tutorial Chair of the International Symposium on Information Theory 2010, the Awards Chair of GLOBECOM 2014, and the General Co-Chair of MMSP'17. He served as an Associate Editor for five IEEE Transactions. He is currently an Associate Editor for the IEEE TRANSACTION ON MULTIMEDIA.