An Efficient Algorithm to Select Reference Views for Virtual View Synthesis

Da-Yoon Nam¹, Woo-Kyung Jung², Hyeon-Deok Han³, and Jong-Ki Han⁴

¹Department of Information and Communication Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, 05006 South Korea
²Department of Information and Communication Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, 05006 South Korea
³Department of Information and Communication Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, 05006 South Korea
⁴Department of Electrical Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, 05006 South Korea

Corresponding author: Jong-Ki Han (e-mail: hjk@sejong.edu).

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) under Grant 2017-0-00486 funded by the Korea government (MSIT).

ABSTRACT

View synthesis is one of the key techniques for generating immersive media. Virtual view synthesis techniques require considerable input views to provide a wide viewing space to users, such as 360 virtual reality. However, the computational complexity is increased, and the synthesized virtual image is blurred when all input views are used as reference views. We analyzed the algorithm complexity and synthesized image quality according to the number of reference pictures. Based on the results, we propose a systematic algorithm to compose an optimal subset of the reference pictures to reduce the hole area and increase the accuracy of the overlapped pixel data in the virtual view. The algorithm consists of the screening and optimal composing steps, which involve logical and geometric measurements. The experimental results demonstrate that the proposed method reduces algorithm complexity and improves the virtual view quality.

INDEX TERMS

Immersive image, reference view selection, virtual view synthesis

I. INTRODUCTION

As applications using immersive media have been developed, various technologies to synthesize virtual views have been studied to construct virtual media systems, such as metaverse, digital twin, and augmented reality (AR) or virtual reality (VR) services, supporting three to six degrees of freedom (DoF). Many companies have developed various virtual media systems with a head-mounted display displaying the screen content. The Moving Picture Expert Group (MPEG), an international standardization organization, formed MPEG-I [1], [2] to standardize the codec for volumetric and immersive media [3]–[9]. In immersive media systems, geometric information, such as camera poses and depth maps, is critical to producing virtual views with high quality. When texture and depth data are provided, a virtual view can be synthesized along an arbitrary viewing angle according to the user’s viewpoint within a limited virtual space.

The methods to synthesize virtual views are classified into the dense light field-based rendering (DLFR) [10], [11] and depth image-based rendering (DIBR) techniques [12]–[24]. In DLFR techniques, multiple views are taken by a set of cameras or micro-lenses, which are arranged with a dense baseline. Optical effects, such as light reflection and transmission, can be restored; thus, these techniques generate realistic, immersive images. However, these methods require professional equipment, such as the plenoptic camera [10]. In addition, to generate content supporting a high DoF, such as 6 DoF, the number of employed cameras increases exponentially, and it can be a burden for developing the systems.

In DIBR techniques [12]–[24], a 6 DoF immersive video can be created using general cameras instead of professional equipment, where the quality of the synthesized view predominantly depends on the quality of the depth map. The depth information resulting from techniques based on DIBR may have a variety of holes, and it degrades the quality of the synthesized image. Therefore, reducing the area of the hole regions and increasing the quality of the depth information are some of the most crucial issues. When depth information is derived, if all neighbor reference views are used to construct a virtual view, using unnecessary and similar reference pictures blurs the synthesized picture. In addition, the computational complexity increases as the number of reference views increases [25]. However, if few reference pictures are used, the necessary information to construct the virtual view is not provided. Thus, the depth map has wide hole regions that
produce distorted images. To derive the depth map, which produces high-quality virtual views, we should select the optimal reference views, maximizing the synthesized picture quality.

Some research has been conducted in the research field on selecting reference pictures [29]–[33] and is explained in detail in Section II. Because the methods have limitations in improving image quality and reducing complexity, we propose an efficient algorithm to optimize the set of reference views based on the cost function, considering the hole size and diversity of the reference pictures used.

This paper is organized as follows. In Section II, we explain the existing view synthesizers and conventional algorithms to select reference views. Section III provides the preliminary analysis for view selection, where we analyze the algorithm complexity and synthesized image quality according to the number of reference views used. These data produce the tendency of the algorithm performance for various scenarios. Based on these data, we design a cost function to represent the algorithm performance and propose an algorithm to optimize the reference pictures used in Section IV. The simulation results are presented to demonstrate the performance of the proposed algorithm in Section V. Finally, we conclude this paper in Section VI.

II. RELATED WORK

Fig. 1 represents the problem of this study that needs to be solved. It shows N pictures to synthesize a virtual view. While the location and pose of the virtual camera change, a synthesized picture is constructed using the related reference pictures. If all reference pictures are used to construct a virtual image, it would be computationally expensive, and the synthesized image would be blurred due to the misalignment of several overlapped image patches. However, if the virtual view is derived with too few reference pictures, the synthesized picture includes many holes due to a shortage of information. Therefore, it is important to select the optimal set of reference pictures out of all input images. The proposed method refers to only the most suitable views to synthesize a virtual view.

Fig. 2 explains the core technologies to synthesize a virtual view, where DIBR methods derive the depth information from the neighbor reference views and use it to construct a virtual view. The DIBR methods were imported into a variety of standard visual systems, such as three-dimensional (3D) television [26], free-viewpoint television [27], and multiview video coding [28], which MPEG made.

FIGURE 1. Conceptual overview diagram: (a) view synthesis without view selection, (b) view synthesis with view selection.

FIGURE 2. Virtual view synthesis algorithm based on DIBR.

The depth image is derived from the selected reference views; thus, selecting the reference views used to construct the depth map among all available reference views is one of the most critical issues. As indicated in the literature, depth maps may include various holes that degrade the quality of the synthesized image because the synthesized image quality...
predominantly depends on the accuracy of the derived depth data. This section explains various conventional methods for view synthesis and reference view selection.

The DIBR algorithms are classified as single view-based [13]–[16], stereo view-based [17]–[19], and multiview-based syntheses [20]–[24] according to the number of the reference views to construct a target virtual view. In the single view-based algorithm category, a single neighbor view is used as a reference view, where the texture image, depth image, and set of camera parameters of the selected neighbor view are used to construct a target virtual view. Because the information for the single-neighbor view is insufficient to generate all depth values in the derived depth image for the target view, the depth image may include a variety of holes that degrade the synthesized picture quality. Various algorithms to remove the holes are proposed in [13]–[16].

In the stereo view-based systems, the left and right views of the target virtual view are used as reference views. Because the amount of information used in the stereo view-based system exceeds that of the single view-based algorithm, the synthesized picture quality from the stereo view-based DIBR is better than that from the single view-based algorithm. Among the stereo view-based algorithms, the technique proposed by Tanimoto et al. [18] was adopted for view synthesis reference software (VSRS) [18], which was created by the 3D video group at MPEG. The VSRS constructs the target view using data from one or two reference views. The picture resulting from the VSRS may have some small holes. Thus, some post-processing to conceal those holes is needed. Zhu et al. [19] proposed a novel algorithm to fill the holes, in which occluded, unoccluded, and invisible backgrounds are identified to fill the holes.

In the category of the multiview-based algorithms, Li et al. [20] proposed a view synthesis framework to exploit the data for multiple reference views to construct a target virtual view. They defined complementary views that are mutually cooperative to reduce the hole sizes in the generated virtual view. In 2018, the MPEG-I visual group published a reference synthesizer that imports a versatile view synthesizer [21], reference view synthesizer (RVS) [22], and view weighting synthesizer [23]. The tools in [21]–[23] construct a virtual view using multiple reference views.

As the number of the reference views increases in these three categories, the computational complexity also increases. As for the synthesized picture quality, when the number of reference views is too small, the synthesized picture may have holes that degrade the virtual view quality. However, when the number of reference views is too large, some parts of the virtual view are constructed from overlapped patches warped from multiple reference views. The overlapped patches can be misaligned, resulting in blurred parts of the synthesized image. Thus, the number of reference views should be optimized to improve the image quality while considering the algorithm complexity.

Various techniques [29]–[33] to select the set of the used reference views have been studied. Maugey et al. [29], [30] proposed a novel reference view selection algorithm for multiview video coding, where coding efficiency and transmission speed are considered in the cost function for optimal selection. However, these techniques are constrained in improving the performance of virtual view synthesizers because they do not consider the variation of the synthesized image quality according to the reference view composition. In [31], the virtual view synthesis (VSVS) algorithm was proposed to select reference views. It selects two reference views that are nearest to the virtual view. The constraint on the number of the reference views to two simplifies the algorithm, but the resulting picture may have various sized holes.

In [32], a 3D photorealistic environment simulator (PreSim) was proposed to select the reference views, where the nearest reference views from the virtual view are selected up to 10 pictures. PreSim excludes some inappropriate reference views. The algorithm assumes that the baselines for all input cameras are dense enough. As we observe from preliminary tests, the algorithm usually selects 10 reference views, although all 10 reference views are unnecessary to increase the synthesized picture quality. With 10 reference views, the algorithm efficiency decreases with respect to the visual quality and computational complexity.

In [33], we proposed an algorithm to compose a set of reference views, where simple conditions were checked to determine whether the corresponding reference view was used. In this research [33], the depth data were not used, and the overlapped regions warped from the reference views were estimated heuristically without a cost function. The simulation results in [33] demonstrated that the algorithm has some constraints in improving the quality of the synthesized image, although it is very simple. In this paper, we estimate the hole size generated by stitching the pictures warped from the neighbor reference views to overcome those problems. In addition, the redundancy among these reference views to synthesize the virtual view is predicted based on the geometric relationship between them. The set of reference views is optimally selected by minimizing the cost function, which is based on the estimated hole size, predicted redundancy, and mutual supplementation between reference views.

III. PRELIMINARY ANALYSIS FOR VIEW SELECTION

In this section, we check the tendency of the performance of the conventional synthesis algorithm concerning the algorithm complexity and virtual view quality according to the number of reference views. The RVS [22] is used as a synthesis algorithm in these tests because it includes a function to set the number of reference views and has excellent performance on natural scene and computer graphic data. Based on the analysis resulting from these preliminary tests, we design the algorithm to select the reference views. Fig. 3 illustrates the four sets (‘umbrella,’ ‘chair,’ ‘checkerboard,’ and ‘C908’) of the test images used in these tests. The resolutions of the
umbrella, chair, checkerboard, and C908 sets are 1920 × 1080, 1280 × 720, 3000 × 4000, and 3000 × 4000, respectively. Fig. 3(d) presents a panoramic image stitched from multiple input pictures.

A. COMPLEXITY ANALYSIS OF THE VIEW SYNTHESIS

Figs. 4 and 5 reveal the CPU time to synthesize a virtual view according to the resolution and number of reference pictures, where computational complexity is approximately proportional to the number and resolution of the reference views. This observation can be represented as follows:

\[ C \approx \alpha \times (n(\text{views}) \times n(\text{pixels})) \]  

where \( C \) and \( \alpha \) are the computational complexity and proportional constant, respectively. The number of reference views is denoted by \( n(\text{views}) \), and \( n(\text{pixels}) \) is the resolution of each reference picture.

B. ANALYSIS FOR HOLE SIZES IN THE SYNTHESIZED IMAGE

The holes resulting from the view synthesis algorithm are classified as ‘uncovered’ and ‘disoccluded’ holes. The uncovered hole is made when the field of view of the virtual view is mismatched with that of the reference view. When some part of the virtual view corresponds to the hidden area of the reference view, the part results in a disoccluded hole. Fig. 6 presents the ratio between the sizes of the holes and images in the synthesized image when the number of reference views varies. In this experiment, the hole size rapidly decreases as the number of reference views increases when the number is small. However, the ratios were saturated over a threshold number of reference views, implying that too many reference views are not needed to reduce the area of holes.
C. QUALITY ANALYSIS FOR THE SYNTHESIZED IMAGE

Fig. 7 depicts the relationship between the synthesized virtual view quality and number of reference views. When the number starts small, as it increases, the peak signal-to-noise ratio (PSNR) of the synthesized image increases because using more reference views provides more data to construct the virtual view. However, the quality is saturated when the number exceeds a specific value, and if the number becomes too large, the PSNR decreases because many patches warped from multiple reference views are misaligned, resulting in blurred regions in the virtual view.

![Figure 7](image)

**FIGURE 7.** Objective quality of the synthesized image according to the number of reference views when RVVS is applied to the test image sets.

D. SUMMARY OF PRELIMINARY ANALYSIS

As we observe from the preliminary analysis, incrementing the number of reference views when RVVS is applied to the test image sets.

![Figure 8](image)

**FIGURE 8.** Estimation of the overlapped region between the virtual view and picture warped from a neighbor reference view.

Initially, all points of \( \{ p_{it}^a, a = tl, tr, bl, br \} \) are projected onto the world coordinate system, respectively, as follows:

\[
\begin{bmatrix}
  x(p_i^a) \\
  y(p_i^a) \\
  z(p_i^a)
\end{bmatrix} = d_i^a \begin{bmatrix}
  1 & 0 & -p_z & \frac{1}{f_x} \\
  0 & 1 & -p_z & \frac{1}{f_y} \\
  0 & 0 & 1 & w(p_i^a)
\end{bmatrix},
\]

where \((x(p_i^a), y(p_i^a), z(p_i^a))\) is the world coordinate for \((u(p_i^a), w(p_i^a))\). The focal lengths \(f_x\) and \(f_y\) and the principal point \((p_x, p_y)\) are intrinsic camera parameters given as input data, and \(d_i^a\) is a depth value of \(p_i^a, a = tl, tr, bl, br\).

After applying (2), \((x(p_i^a), y(p_i^a), z(p_i^a))\) is rotated and translated to the corresponding coordinate \((x_v(p_i^a), y_v(p_i^a), z_v(p_i^a))\) in the virtual camera coordinate system as follows:

\[
\begin{bmatrix}
  x_v(p_i^a) \\
  y_v(p_i^a) \\
  z_v(p_i^a)
\end{bmatrix} = R_{t \rightarrow v} x_v(p_i^a) + t_{t \rightarrow v},
\]

where \(R_{t \rightarrow v}\) and \(t_{t \rightarrow v}\) are the rotation matrix and translation vector from the coordinate system of the \(i\)th camera to that of the virtual camera, respectively. In addition, \(R_{t \rightarrow v}\) and \(t_{t \rightarrow v}\) are calculated using

\[
R_{t \rightarrow v} = R_t^T R_v,
\]

\[
t_{t \rightarrow v} = -R_t^T (t_v - t_t),
\]

where \(R_t\) and \(t_t\), and \(R_v\) and \(t_v\) are the rotation matrices and translation vectors of the \(i\)th and virtual views, respectively. The superscript \(T\) indicates the transpose of the matrix. Then, \((x_v(p_i^a), y_v(p_i^a), z_v(p_i^a))\) from (3) is reprojected onto the coordinate system in the virtual view using the following equation:

\[
d_i^a \times \begin{bmatrix}
  u(p_i^a) \\
  w(p_i^a)
\end{bmatrix} = K_v \begin{bmatrix}
  x_v(p_i^a) \\
  y_v(p_i^a) \\
  z_v(p_i^a)
\end{bmatrix},
\]

where \(d_i^v\) and \((u(p_i^a), w(p_i^a))\) are the depth and coordinate of the reprojected point \(p_i^a\) for \(a = tl, tr, bl, br\) respectively, and \(K_v\) is the intrinsic matrix of the virtual camera. Using (6) provides \(d_i^v\) and \((u(p_i^a), w(p_i^a))\), where \(a = tl, tr, bl, br\). 

IV. PROPOSED ALGORITHM

A. ESTIMATION OF OVERLAP REGION

Fig. 8 presents a scenario where the \(i\)th reference picture is warped to construct a virtual view, where the top-left, top-right, bottom-left, and bottom-right corners of the \(i\)th reference picture are denoted by \(p_{it}, p_{ir}, p_{bl}, p_{br}\), respectively. We define the coordinates of the four corner points \(\{ p_{it}, p_{ir}, p_{bl}, p_{br}\} \) by \(\{ (u(p_{i1}^a), w(p_{i1}^a)), a = tl, tr, bl, br \} \). The following processes are applied to derive the point coordinates for \(\{ p_{it}, p_{ir}, p_{br}, p_{bl}\} \), which are warped from \(\{ p_{it}, p_{ir}, p_{bl}, p_{br}\} \).
If $d_{ai}'$ is a negative value, it implies that the point $p_{ai}'$ is located behind the center of the virtual camera in the 3D world coordinate system. Thus, the reference picture whose warped depth $d_{ai}''$ satisfies the following equation is excluded from the set of reference views to synthesize the virtual view:

$$d_{ai}'' < 0, \ a = tl, tr, bl, br.$$  \hspace{1cm} (7)

In Fig. 8, if any warped corners are out of the virtual view range, they are mapped to the closest corner points $\{p_{ai}''\} (a = tl, tr, bl, br)$ within the virtual view. When the warped corner is in the virtual view, $p_{ai}''$ equals $p_{ai}'$.

The area $A(OR^i)$ of the region $OR^i$ overlapped by the warped $i^{th}$ reference picture is calculated using Heron’s formula as follows:

$$A(OR^i) = \frac{\sqrt{(u(p_{ai}''') - u(p_{ai}''))(w(p_{ai}''') - w(p_{ai}'')) + (u(p_{ai}''') - u(p_{ai}''))(w(p_{ai}''') - w(p_{ai}'')) - (u(p_{ai}''') - u(p_{ai}''))(w(p_{ai}''') - w(p_{ai}''))}}{2},$$ \hspace{1cm} (8)

where $(u(p_{ai}'''), w(p_{ai}''''))$ are the 2D coordinates of $p_{ai}'''$ for $a = tl, tr, bl, br$, respectively. If (8) produces a negative value of $A(OR^i)$, the rectangle comprising $\{p_{ai}^{tl}, p_{ai}^{tr}, p_{ai}^{bl}, p_{ai}^{br}\}$ is flipped and twisted. Thus, if the following condition is satisfied, the corresponding $i^{th}$ reference picture is not considered in the synthesis:

$$A(OR^i) < 0.$$ \hspace{1cm} (9)

### B. ESTIMATION FOR DISOCCLUSION

In this section, we analyze a scenario generating a disocclusion and design a scheme to measure the disocclusion quantity. Fig. 9 displays the scenario to generate disocclusion, where a blue column is in front of a big gray box. Some points in the gray box cannot be captured by the $i^{th}$ camera because the blue column blocks them. In this scenario, the reference picture captured by the $i^{th}$ camera may produce a kind of disocclusion in the synthesized image.

In Fig. 9, we assume that the virtual and $i^{th}$ cameras examine the same point $p_{ci}'$, where $c_v$ and $c_i$ are the centers of the virtual and $i^{th}$ cameras. We draw two rays from the centers of the two cameras to the back object through the point $p_{ci}'$. The points $p_{ri}$ and $p_{ri}$ are intersections between the rays and rear object. In addition, $g^i$ is a vector from $p_r$ to $p_{ri}$, whose absolute value $|g^i|$ is highly correlated with the hole size in the synthesized picture. Additionally, $d_f$ is the farthest depth of the camera and is given as the input metadata. Finally, $d_i$ is the depth of $p_{ci}'$ in the coordinate system for the virtual camera.

In Fig. 9, $c_i$ can be represented with a relative position from the center $c_v$ of the virtual camera (i.e., $c_i$ and $c_v$ are represented as $(sx, sy, sz)^T$ and $(0, 0, 0)^T$, respectively). The position of $c_i$ is calculated using extrinsic parameters for the input and virtual cameras:

$$c_i = -R^T_x(t_x - t_i).$$ \hspace{1cm} (10)

From the geometric relation in Fig. 9, $\overrightarrow{p_{ci}'p_{ri}}$ and $\overrightarrow{p_{ri}p_{ri}}$, and $g_i^i$ are derived as follows:

$$\overrightarrow{p_{ci}'p_{ri}} = \frac{d_f - d_i}{d_i - c_i} \overrightarrow{p_{ci}'},$$ \hspace{1cm} (11)

$$\overrightarrow{p_{ri}p_{ri}} = \frac{d_f - d_i}{d_i - c_i} \overrightarrow{p_{ci}'},$$ \hspace{1cm} (12)

$$g_i^i = \overrightarrow{p_{ci}'p_{ri}} - \overrightarrow{p_i}p_r.$$ \hspace{1cm} (13)

To analyze the relation between $g_i^i$ and the hole size in the virtual image, we present Fig. 10, the top view of Fig. 9, where $g_i^i$ is mapped to the vector $\overrightarrow{l'}$ in the virtual image plane as follows:

$$\overrightarrow{l'} = \begin{bmatrix} f \ 0 \ 0 \\ 0 \ f_d \ 0 \end{bmatrix} \overrightarrow{g_i^i}. \hspace{1cm} (14)$$

The components of vector $\overrightarrow{l'}$ are denoted by $\left( x \left( \overrightarrow{l'} \right), y \left( \overrightarrow{l'} \right) \right)^T$. 

---

**FIGURE 9.** Scenario generating disocclusion.

**FIGURE 10.** Top view of Fig. 9.
If the magnitude $|\vec{t}|$ of the vector is large, we can also predict that the hole size is large. Thus, if both $|x(\vec{t})|$ and $|y(\vec{t})|$ are larger than their thresholds of $Th_x$ and $Th_y$, respectively, then the $i^{th}$ reference picture is not used to synthesize the virtual view:

$$\left| x(\vec{t}) \right| > Th_x$$  \hspace{1cm} (15)

$$\left| y(\vec{t}) \right| > Th_y$$  \hspace{1cm} (16)

### C. REMOVAL OF REDUNDANT REFERENCE PICTURES

Fig. 11 provides an example of redundant reference views, where $OR^i$ and $OR^j$ are generated from the warping of the $i^{th}$ and the $j^{th}$ reference pictures $I_i$ and $I_j$, respectively. The $OR^i$ and $OR^j$ are estimated using the method in Section IV-A.

![Figure 11. Example of redundant reference views: (a) redundant overlap regions and (b) redundant reference views.](image)

In Fig. 11 (a), if $OR^j$ is included completely in $OR^i$ (i.e., $OR^j \subset OR^i$), the $j^{th}$ reference picture $I_j$ may be redundant for $I_i$. This redundancy can be checked using the following four equations:

$$w(p'_{jx}) - w(p'_{ix}) - u(p'_{jy}) + u(p'_{iy}) + w(p'_{jy}) - u(p'_{ix}) - u(p'_{iy}) < w(p'_x)$$  \hspace{1cm} (17)

$$w(p_{jx}) - w(p_{ix}) - u(p_{jy}) + u(p_{iy}) + w(p_{jy}) - u(p_{ix}) - u(p_{iy}) < w(p_x)$$  \hspace{1cm} (18)

$$u(p'_{jx}) - u(p'_{ix}) - w(p'_{jy}) + w(p'_{iy}) + u(p'_{jy}) - w(p'_y) + w(p'_y) < u(p'_x)$$  \hspace{1cm} (19)

$$u(p_{jx}) - u(p_{ix}) - w(p_{jy}) + w(p_{iy}) + u(p_{jy}) - w(p_y) + w(p_y) < u(p_x)$$  \hspace{1cm} (20)

where $a$ is one of $\{t_l, t_r, b_l, b_r\}$, and $(u(p'_{ix}'), u(p'_{iy}'))$, $(u(p_{ix}'), u(p_{iy}'))$, $(u(p'_{ix}'), w(p_{iy}'))$, $(u(p_{ix}'), w(p_{iy}'))$, and $(u(p'_{ix}'), (u(p_{ix}'), w(p_{iy}'))$ are the coordinates of the four corners $p'_{i(x)}$, $p_{i(x)}$, $p'_{i(y)}$, $p_{i(y)}$ for the $i^{th}$ overlap region $OR^i$. In addition, $(u(p'_x), w(p'_y))$ is a coordinate of one of the four corners for the $j^{th}$ overlap region $OR^j$, where $a$ is set to one among $\{t_l, t_r, b_l, b_r\}$. If Equations (17) to (20) are all satisfied with $a = \{t_l, t_r, b_l, b_r\}$, then $OR^j$ is included completely in $OR^i$ (i.e., $OR^j \subset OR^i$).

In Fig. 11, if $OR^j$ includes some holes, the $OR^j$ has wider holes because the difference between the shooting angles of the virtual and $j^{th}$ cameras is larger than that between the virtual and the $i^{th}$ cameras. This difference can be checked using the following equations:

$$(x(\vec{t}) \geq x(\vec{t}) \geq 0) \lor (x(\vec{t}) \leq x(\vec{t}) \leq 0)$$  \hspace{1cm} (21)

$$(y(\vec{t}) \geq y(\vec{t}) \geq 0) \lor (y(\vec{t}) \leq y(\vec{t}) \leq 0)$$  \hspace{1cm} (22)

In the algorithm, if all conditions for (17) to (22) are satisfied, the $j^{th}$ reference view is excluded in the set of reference pictures.

### D. PROPOSED ALGORITHM FOR REFERENCE VIEW SELECTION

When $N$ reference views are given to synthesize a virtual view as in Fig. 2, each reference picture is checked regarding whether it is excluded from the set of reference images based on the criteria of (9), (15)–(16), and (17)–(22). After screening them, the remaining reference pictures are included in the universal set $B$ of the candidate reference images. In this section, we propose an algorithm to construct the optimal set $S^*$ from the universal set $B$, where $S^* \subset B$. The set $S^*$ is optimized based on the cost function considering the estimated hole size, predicted redundancy, and mutual supplementation between reference views. We summarize the notations consisting of the cost function in Table I.

| Notation | Meaning |
|----------|---------|
| $B$ | Universal set of candidate reference views |
| $r_k$ | $k^{th}$ reference view included in set $B$, $r_k \in B$ |
| $n_{min}$ | Target minimum number of reference views for selection |
| $M$ | Number of the views included in set $B$ |
| $OR^k$ | Overlap region warped from reference view $r_k$ |
| $\vec{k}$ | Direction vector resulting from reference view $r_k$ |
| $x(\vec{k})$ | $x$ component of $\vec{k}$ |
| $y(\vec{k})$ | $y$ component of $\vec{k}$ |
| $p'^c_k$ | Center position in the overlap region $OR^k$ |
| $x(p'^c_k)$ | $x$ component in the coordinate of $p'^c_k$ |
| $y(p'^c_k)$ | $y$ component in the coordinate of $p'^c_k$ |
| $H^k_{or}$ | Height of $OR^k$ |
| $W^k_{or}$ | Width of $OR^k$ |
| $H_{img}$ | Height of virtual view image |
| $W_{img}$ | Width of virtual view image |
| $S_Z$ | Subset of set $B$ (the number of reference views in $S_Z$ is $Z$) |
| $Z$ | Number of reference views in set $S_Z$ |
| $S^*$ | Specific set minimizing the cost function among the available subsets ($S_Z$) |

### TABLE I. DEFINITIONS FOR NOTATION

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
The cost function in the proposed algorithm is represented as follows:

\[
C_T(S_Z) = \begin{cases} 
C_h(S_Z) & \text{if } Z = 1, \\
C_h(S_Z) + \lambda T C_d(S_Z) + \lambda T C_T(S_Z) & \text{else},
\end{cases}
\]  

(23)

where the lower subscripts \(T, H, D, \) and \(V\) of \(C_T(S_Z), C_H(S_Z), \) \(C_D(S_Z)\), and \(C_V(S_Z)\) correspond to “Total,” “Hole,” “Direction,” and “Variance,” respectively. \(S_Z\) denotes a specific subset of the reference pictures. The number of pictures in \(S_Z\) is \(Z\). Table I defines the notations used to calculate the cost terms in this section.

In (23), \(C_H(S_Z)\) is an estimator for the average area of the total holes generated using the reference pictures in \(S_Z\). To calculate \(C_H(S_Z)\), the averaged noncovered and disocclusion holes are estimated using \(C_{\text{un}}(S_Z)\) and \(C_{\text{dis}}(S_Z)\), respectively, as follows:

\[
C_H(S_Z) = C_{\text{un}}(S_Z) + C_{\text{dis}}(S_Z),
\]

(24)

where

\[
C_{\text{un}}(S_Z) = \frac{1}{Z} \sum_{n \in S_Z} \frac{H_{\text{img}} W_{\text{img}} - OR^k}{H_{\text{img}} W_{\text{img}}},
\]

(25)

\[
C_{\text{dis}}(S_Z) = \frac{1}{Z} \sum_{n \in S_Z} \frac{H_{\text{img}} W_{\text{img}}}{H_{\text{img}} W_{\text{img}}} \left[ x \left( \frac{p_k}{W_{\text{img}}} \right) + y \left( \frac{p_k}{W_{\text{img}}} \right) \right],
\]

(26)

In (25) and (26), the geometric meanings of \(x \left( \frac{p_k}{W_{\text{img}}} \right)\) and \(y \left( \frac{p_k}{W_{\text{img}}} \right)\) are shown in both of Fig. 9 and Fig. 10.

In the scenario in Fig. 2, if the reference pictures \{\(r_{\text{left}}, r_{\text{right}}\}\) on the left and right sides of the virtual view are used, they can complementarily remove the disocclusion holes in the synthesized picture. To verify this case, we calculate the absolute value of the sum of \(p_k^r\) for all views in \(S_Z\). A smaller absolute value results in a smaller averaged area of holes. Nevertheless, the average area of the holes decreases as the absolute value of the sum of the center position \(p_k^c\) for all reference views in \(S_Z\) decreases. The geometric meaning of \(p_k^c\) is shown in Fig. 9 and Fig. 10. The terms related to the absolute values of the sum of \(p_k^r\) and that of the center position \(p_k^c\) are calculated in (27) and (28), respectively, as follows:

\[
C_p(S_Z) = \frac{1}{Z} \left[ \sum_{n \in S_Z} \frac{x(p_k^c)}{W_{\text{img}}} + \frac{1}{2} \sum_{n \in S_Z} \frac{y(p_k^c)}{H_{\text{img}}} \right],
\]

(27)

\[
C_p(S_Z) = \frac{1}{Z} \left[ \sum_{n \in S_Z} \frac{x(p_k^c)}{W_{\text{img}}} - \frac{1}{2} \sum_{n \in S_Z} \frac{y(p_k^c)}{H_{\text{img}}} \right],
\]

(28)

where (27) and (28) comprise the direction-related term \(C_p(S_Z)\) given in (23):

\[
C_p(S_Z) = C_h(S_Z) + C_p(S_Z)
\]

(29)

The direction term \(C_D(S_Z)\) is related to the relative directions and positions of the reference pictures used to construct a virtual view. Minimization of \(C_D(S_Z)\) reduces hole size in the virtual view.

In (23), \(C_V(S_Z)\) represents the uniformity of the positions of the reference pictures. If most reference views are located in a specific area, some would be redundant for reducing the hole area. We check the direction vector \(\vec{f}_k\) for each view \(r_k\) in \(S_Z\) to consider the uniformity, as follows:

\[
\theta^k = \begin{cases} 
\tan^{-1} \left( \frac{y(\vec{f}_k)}{x(\vec{f}_k)} \right), & \text{if } x(\vec{f}_k) \geq 0, \\
-\tan^{-1} \left( \frac{y(\vec{f}_k)}{x(\vec{f}_k)} \right), & \text{else if } x(\vec{f}_k) < 0.
\end{cases}
\]

(30)

Based on (30), we calculate the absolute difference \(\Delta \theta^k\) for all views in \(S_Z\) as follows:

\[
C_V(S_Z) = \frac{1}{Z} \sum_{n \in S_Z} \left( \frac{1}{2\pi} \left( \Delta \theta^k - \frac{2\pi}{Z} \right) \right)^2.
\]

(32)

In (32), \(2\pi/Z\) represents the average value of \(\Delta \theta^k\) in an ideal scenario, where the reference pictures are located uniformly with the angle intervals of \(2\pi/Z\).

**TABLE II. PSEUDOCODE FOR THE PROPOSED ALGORITHM**

**Iterative algorithm to select reference pictures**

**Input:** \(B\) is the universal set of remaining view candidates \(n_{\text{min}}\) is the minimum number of reference views

**Initialization:** Set \(Z = 1, S_0, S_0^*, S^* = \emptyset\)

**Output:** Optimal reference view set \(S^*\)

While \(Z \leq M\) do:

For each view \(r_k \in (B - S_{Z-1})\)

Update subset \(S_Z = r_k \cup S_{Z-1}\)

Compute \(C_T(S_Z)\)

If \(C_T(S_Z) > C_T(S_Z^*)\)

Update \(S_Z^* = S_Z\)

End if

End for

If \((Z > n_{\text{min}}\) and \(C_T(S_Z^*) > C_T(S^*)\))

Break

End if

Update \(S^* = S_Z^*\)

\(Z = Z + 1\)

End While
In (23), we only evaluate \( C_T(S_Z) \) when only one reference view was included in a subset \( S_Z \) (i.e., \( Z = 1 \)), because the statistics, \( C_H(S_Z) \) and \( C_D(S_Z) \), of multiple reference views cannot be calculated. Based on the cost function of (23), we derive the optimal reference view set \( S^* \) that minimizes the total function \( C_T(S_Z) \). Calculating the cost functions for all constructible sets \( S_Z \), 1 ≤ \( Z \) ≤ \( M \), is highly complex. Therefore, we optimize the set based on the greedy algorithm [34], which determines the optimal solution by considering the best available choice at every iteration.

We determine a set \( S_Z^* \), minimizing the cost function for a specific value of \( Z \); after that, we increase \( Z \) at the end of each iteration. Initially, \( S_0, S_0^* \), and \( S^* \) are set to empty, and \( C_T(S_0) \) is set as the positive numerical limit. The initial value of \( Z \) is set to 1. The proposed algorithm is represented with pseudocode, as shown in Table II.

### E. OPTIMIZATION OF COEFFICIENTS

When the total cost function \( C_T(S_Z) \) in (23) is calculated, coefficients \( \lambda_1 \) and \( \lambda_2 \) are used to assign differential weight factors for \( C_H(S_Z) \), \( C_D(S_Z) \), and \( C_V(S_Z) \). To optimize the factors, we analyzed the effects of \( C_H(S_Z) \), \( C_D(S_Z) \), and \( C_V(S_Z) \) for the synthesized picture quality through empirical tests. Fig. 12 illustrates the relationship between the PSNR of the synthesized picture and the values for each term of (23), where 16 test image sets were used. Fig. 12 indicates linear correlations in all tests. Moreover, the PSNR decreases as the values of \( C_H(S_Z) \), \( C_D(S_Z) \), and \( C_V(S_Z) \) increase, implying that the terms in (23) efficiently represent the cost to create the synthesized picture.

Considering the linear dependency in Fig. 12, the coefficients \( \lambda_1 \) and \( \lambda_2 \) are set to the following:

\[
\lambda_1 = \frac{\Delta \text{psnr}}{\Delta C_H(S_Z)} \approx 1.0, \tag{33}
\]

\[
\lambda_2 = \frac{\Delta \text{psnr}}{\Delta C_V(S_Z)} \approx 1.5, \tag{34}
\]

where \( \frac{\Delta \text{psnr}}{\Delta C_H(S_Z)} \), \( \frac{\Delta \text{psnr}}{\Delta C_D(S_Z)} \), and \( \frac{\Delta \text{psnr}}{\Delta C_V(S_Z)} \) are the slope values of the trend line between the PSNR and \( C_H(S_Z) \), \( C_D(S_Z) \), and \( C_V(S_Z) \), respectively.

### F. CONTRIBUTION

This paper proposes a systematic algorithm to compose an optimal subset of the reference pictures to reduce the hole area and increase the accuracy of the overlapped pixel data in the virtual picture. The process consists of (step 1) screening the redundant reference pictures and (step 2) composing the optimal subset from the remained reference pictures. These steps are conducted based on logical and geometric measurements.

In the screening step, all reference pictures are tested according to (A) the area of the overlap region, (B) disocclusion, and (C) redundancy; these terms are measured with geometric metrics, as explained in Section IV. A, B, and C, respectively.

After screening, it is challenging to compose the optimal subset of the reference pictures out of the remained \( M \) reference pictures, where the number of reference pictures in the optimal subset can be in the range of 1–M. In Table II, we propose an algorithm to compose the subset of the reference pictures, where the number of the reference pictures in the subset increases from 1 to \( M \) as the process repeats iteratively. In the composing step, (a) hole area, (b) the complementary relationship of the reference pictures, and (c) uniformity of

**FIGURE 12.** Dependencies between one of three costs \( C_H(S_Z) \), \( C_D(S_Z) \), and \( C_V(S_Z) \), and the PSNR of the synthesized image.
positions of reference pictures are evaluated by the logical cost function based on equations (23)–(32).

Using the selected reference pictures only, instead of all reference pictures, significantly reduces the computational complexity, because decreasing the number of the used reference pictures reduces the number of very complex modules, such as the depth information generation, warping, blending, and post-processing for the used reference pictures. It means that the proposed algorithm is effective in reducing computational complexity.

On the other hand, as for the quality of the synthesized picture, it is obvious that the degradation of the virtual view dominantly depends on the hole size. The terms in (15), (16), and (24), which are related to the hole size, are used to check whether the reference picture is appropriate to construct the virtual view minimizing the hole size. We can expect that the hole size is reduced in the synthesized picture because the screening and composing of the optimal subset are based on those equations.

Additionally, the terms in (17)–(22), (29), and (32) are related to the mutual compensation of the reference pictures. The compensation increases the quality of the synthesized picture. Therefore, we can expect that the proposed algorithm is effective in increasing the quality of the virtual view.

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

We compared the proposed method with various latest conventional methods, such as VSVS [31], PreSim [32], and no-selector to demonstrate the performance of the proposed algorithm. When no-selector is used in the synthesis, all reference views are employed to create a virtual view. VSVS and PreSim were constructed using the C++ language while implementing these algorithms. The proposed algorithm and various conventional methods are used as one module in the entire system for virtual view synthesis to select the reference views. In this simulation, RVS was used as the view synthesis system because it is a state-of-the-art virtual view synthesizer. The RVS is a part of the MPEG-I reference software [22]. The software for RVS was implemented using C++ and the OpenGL Shading Language. Therefore, the programs were executed on a graphics processing unit. Furthermore, all experiments were executed on a computer equipped with an AMD Ryzen processor at 3.40 GHz with 32 GB RAM and NVIDIA GeForce RTX 2070 SUPER.

The algorithms are tested with various picture sets shown in Figs. 3, 13, and 14. The authors took the pictures in Figs. 3 and 13. Figs. 3 and 13 show each umbrella, chair, checkerboard, C908, checkerboard2, sink, animal, cabinet, wallpaper, wallpaper2 image set. These picture sets were taken using the built-in smartphone camera in Xiaomi Redmi Note 8T. The camera resolution is 3000 \times 4000. The baseline between adjacent cameras is approximately 15 cm to 1 m. All datasets were captured casually with arbitrary movement. The information on the testing sets is summarized in Table III.

The pictures in Fig. 14 are MPEG-I test sequences used under common test conditions during the standardization of the immersive video codec. Fig. 14 depicts each picture: a barn, breakfast, breaktime, frog, Magritte, and mirror. The information on the testing sets is summarized in Table IV.
| Testing Set | Resolution | # Input View | Format  |
|-------------|------------|--------------|---------|
| Umbrella    | 1920 x 1080| 8            | YUV 4:2:0 |
| Chair       | 1280 x 720 | 16           | YUV 4:2:0 |
| Checkerboard| 3000 x 4000| 16           | YUV 4:2:0 |
| Checkerboard2| 3000 x 4000| 31          | YUV 4:2:0 |
| Sink        | 3000 x 4000| 31           | YUV 4:2:0 |
| Stuffed animal | 3000 x 4000| 31        | YUV 4:2:0 |
| Cabinet     | 3000 x 4000| 31           | YUV 4:2:0 |
| Wallpaper   | 3000 x 4000| 31           | YUV 4:2:0 |
| Wallpaper2  | 3000 x 4000| 31           | YUV 4:2:0 |

| TABLE IV. INFORMATION FOR MPEG-I TEST IMAGE SETS IN FIG. 14 |
|------------------------------------------------------------|
| Testing Set      | Resolution   | # Input View | Format  |
| Barn            | 1920 x 1080  | 15           | YUV 4:2:0 |
| Breakfast       | 1920 x 1080  | 15           | YUV 4:2:0 |
| Breaktime       | 1920 x 1080  | 23           | YUV 4:2:0 |
| Frog            | 1920 x 1080  | 13           | YUV 4:2:0 |
| Magritte        | 2000 x 2000  | 46           | YUV 4:2:0 |
| Mirror          | 1920 x 1080  | 15           | YUV 4:2:0 |

| TABLE V. PEAK SIGNAL-TO-NOISE RATIOS (dB) OF SYNTHESIZED PICTURES FOR TEST IMAGE SETS IN FIGS. 3 AND 13 |
|--------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Proposed Algorithm |
| Umbrella     | 23.431      | 23.005| 23.431 | 23.014 |
| Chair        | 23.940      | 24.293| 24.352 | 24.788 |
| Checkerboard | 24.919      | 24.619| 25.236 | 25.273 |
| Checkerboard2| 24.930      | 25.221| 25.762 | 25.862 |
| Sink         | 21.505      | 19.617| 22.080 | 23.814 |
| Animal       | 23.483      | 23.687| 24.348 | 25.049 |
| Cabinet      | 23.576      | 21.282| 24.054 | 24.894 |
| Wallpaper    | 23.798      | 22.949| 24.512 | 26.187 |
| Wallpaper2   | 25.863      | 24.862| 26.451 | 27.071 |
| Average      | 23.938      | 22.282| 24.469 | 25.106 |

| TABLE VI. PEAK SIGNAL-TO-NOISE RATIOS (dB) OF SYNTHESIZED PICTURES FOR MPEG-I TEST IMAGE SETS |
|-----------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim |
| Barn         | 25.565      | 26.181| 26.153 |
| Breakfast    | 29.846      | 29.282| 30.268 |
| Breaktime    | 31.189      | 32.208| 32.580 |
| Frog         | 26.218      | 30.157| 27.247 |
| Magritte     | 30.982      | 31.993| 32.074 |
| Mirror       | 24.523      | 25.284| 25.109 |
| Average      | 28.054      | 29.184| 28.905 |

| TABLE VII. STRUCTURAL SIMILARITY INDEX MEASURE OF SYNTHESIZED PICTURES FOR TEST IMAGE SETS OF FIGS. 3 AND 13 |
|----------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Prop  |
| Umbrella     | 0.796522   | 0.813158| 0.796522| 0.815746 |
| Chair        | 0.697505   | 0.702808| 0.701718| 0.702134 |
| Checkerboard | 0.753287   | 0.744791| 0.760090| 0.760581 |
| Checkerboard2| 0.683382   | 0.681624| 0.698203| 0.698046 |
| Sink         | 0.612609   | 0.575426| 0.618278| 0.638391 |
| Animal       | 0.721330   | 0.739286| 0.749936| 0.770378 |
| Cabinet      | 0.724545   | 0.663028| 0.739555| 0.759081 |
| Wallpaper    | 0.703939   | 0.683163| 0.728647| 0.772943 |
| Wallpaper2   | 0.715030   | 0.679387| 0.723158| 0.732403 |
| Average      | 0.712017   | 0.698075| 0.724012| 0.738856 |

| TABLE VIII. STRUCTURAL SIMILARITY INDEX MEASURE OF SYNTHESIZED PICTURES FOR MPEG-I TEST IMAGE SETS |
|----------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Prop  |
| Barn         | 0.829607   | 0.856716| 0.844898| 0.869323 |
| Breakfast    | 0.915614   | 0.914470| 0.920133| 0.923462 |
| Breaktime    | 0.938332   | 0.940449| 0.946324| 0.947725 |
| Frog         | 0.832940   | 0.902749| 0.855949| 0.896978 |
| Magritte     | 0.925770   | 0.921729| 0.929213| 0.926936 |
| Mirror       | 0.743902   | 0.779378| 0.761325| 0.805399 |
| Average      | 0.864361   | 0.885915| 0.876307| 0.894971 |

| TABLE IX. TIME (SEC/FRAME) FOR ALGORITHMS TO SELECT REFERENCE VIEWS FOR TEST IMAGE SETS OF FIGS. 3 AND 13 |
|------------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Prop |
| Umbrella     | 1.065       | 0.451| 1.063  | 0.532 |
| Chair        | 0.905       | 0.215| 0.641  | 0.227 |
| Checkerboard | 11.453      | 2.513| 8.738  | 3.148 |
| Checkerboard2| 23.587      | 2.114| 7.719  | 3.342 |
| Sink         | 23.594      | 2.513| 7.825  | 3.257 |
| Stuffed animal | 22.04      | 2.784| 7.712  | 3.122 |
| Cabinet      | 21.828      | 3.042| 7.865  | 3.208 |
| Wallpaper    | 22.633      | 2.925| 8.24   | 3.213 |
| Wallpaper2   | 23.136      | 2.897| 8.475  | 2.946 |
FIGURE 15. Images synthesized by various algorithms to select the reference views. The results for umbrella, checkerboard, frog, sink, and animal are depicted from top to bottom. From left to right, no-selector, VSVS, PreSim, and the proposed algorithm are employed in the RVS.
FIGURE 16. A zoomed-in images of highlighted part in images of Fig 15. (a), (b) from the umbrella, (c), (d) from the checkerboard, (e) from the frog, (f), (g) from the sink, (h), (i) from the animal, respectively. From left to right, no-selector, VSVS, PreSim, and the proposed algorithm are employed in the RVS.
B. SUBJECTIVE QUALITY OF SYNTHESIZED IMAGES

In Fig. 15, we compare the visual qualities of the virtual view pictures synthesized by RVS incorporating various reference view selection methods, such as no selector, VSVS, PreSim, and the proposed algorithm, from the left-most column to the right-most column. This figure presents the results for umbrella, checkerboard, frog, sink, and animal from the top to bottom rows. The important regions to compare are indicated with red rectangles.

In the top row, the left rectangle regions of the umbrellas synthesized by no-selector and PreSim are stained because these methods make a virtual view using inappropriate reference pictures. The right rectangle region of the umbrella constructed by the proposed algorithm has sharper edges than in other methods.

In the second row, VSVS makes the stretched parts, resulting from filling the holes. It degrades the visual quality severely. When we compare the images synthesized from the frog data set in the third row, the face of a man in the pictures from no-selector, VSVS, and PreSim are severely blurred, whereas the proposed algorithm provides clear pixel values. In the fourth and fifth rows, VSVS reveals blurred and stretched regions because the method selects only two reference pictures and data are lacking to construct the virtual view results in the stretched regions.

Fig. 16 shows the enlarged figures of the red rectangle regions of Fig. 15, which highlights the visual differences between the pictures in Fig. 15. Based on these results in Figs. 15 and 16, the proposed algorithm outperforms other conventional methods concerning the visual quality of the synthesized image.

C. OBJECTIVE QUALITY OF SYNTHESIZED IMAGES

We calculate the PSNR and structural similarity index measure (SSIM) of the synthesized pictures to evaluate the objective performances of the algorithms. The PSNR and SSIM were measured between synthesized and ground truth images. The averaged values for all views are summarized in Tables V to VIII, where the best and second-best values are red and blue, respectively.

In Table V, the experiments are conducted with test image sets of Figs. 3 and 13, which were taken along arbitrary directions at irregular positions by authors. In this table, the proposed algorithm has the best performance for all test image sets except for the umbrella image set. For the umbrella set, because test pictures were taken within a narrow view angle with a wide baseline width, the best performance was obtained when many reference pictures were used. Therefore, PSNR increases as the number of reference views increases.

The PSNRs for the MPEG-I test data sets are summarized in Table VI, where the proposed algorithm has the best and second-best performances for all data sets. Among the test data sets, the pictures in the frog set were taken by multiple cameras set elaborately. Thus, the depth information for the frog data is exact and provides high-quality data to construct the virtual view. In this case, using two (left and right) reference pictures efficiently synthesizes the virtual view. Thus, VSVS has the best PSNR for the frog data.

The SSIM in Tables VII and VIII implies the structural synthesized picture quality. As we observe from the data in these tables, the proposed algorithm results in the highest and second-highest SSIMs for all test image sets. Based on the results in Tables V to VIII, the proposed algorithm objectively outperforms other methods.

| TABLE X. | TIME (SEC/FRAME) FOR ALGORITHMS TO SELECT REFERENCE VIEWS FOR MPEG-I TEST IMAGE SET |
|----------|---------------------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Prop |
| Barn    | 2.037 | 0.355 | 1.621 | 0.615 |
| Breakfast | 2.08  | 0.357 | 1.540 | 0.622 |
| Breaktime | 3.271 | 0.389 | 1.472 | 0.663 |
| Frog    | 2.032 | 0.471 | 1.484 | 0.443 |
| Magritte | 12.415 | 0.963 | 2.926 | 1.145 |
| Mirror  | 2.280 | 0.602 | 1.655 | 0.562 |

| TABLE XI. | NUMBER OF REFERENCE VIEWS TO SYNTHESIZE A VIRTUAL VIEW FOR TEST IMAGE SETS OF Figs. 3 and 13 |
|------------|-------------------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Prop |
| Umbrella   | 7    | 2    | 7    | 3.250 |
| Chair      | 15   | 2    | 10   | 3.1875 |
| Checkerboard | 15   | 2    | 10   | 3.2500 |
| Checkerboard2 | 31   | 2    | 10   | 3.4686 |
| Sink       | 31   | 2    | 10   | 3.2500 |
| Stuffed animal | 31   | 2    | 10   | 3.1875 |
| Cabinet    | 31   | 2    | 10   | 3.2186 |
| Wallpaper  | 31   | 2    | 10   | 3.2500 |
| Wallpaper2 | 31   | 2    | 10   | 3.1875 |

| TABLE XII. | NUMBER OF REFERENCE VIEWS TO SYNTHESIZE A VIRTUAL VIEW FOR MPEG-I TEST IMAGE SETS |
|------------|-------------------------------------------------------------|
| Testing Data | No-Selector | VSVS | PreSim | Prop |
| Barn    | 14 | 2 | 10 | 3.2000 |
| Breakfast | 14 | 2 | 10 | 3.2000 |
| Breaktime | 22 | 2 | 10 | 3.3913 |
| Frog    | 14 | 2 | 10 | 2.7333 |
| Magritte | 45 | 2 | 10 | 3.3913 |
| Mirror  | 14 | 2 | 10 | 3.2000 |

D. COMPUTATIONAL COMPLEXITIES

To compare the computational complexities of the algorithms, we measured the CPU time consumed by those as summarized.
in Tables IX and X. Table IX lists the results for the test image sets in Figs. 3 and 13. In Table X, the times for the MPEG-I test sets are summarized. The time is represented by sec/frame, the average value for the entire picture in each test data set.

As observed in these tables, VSVS is the fastest algorithm for all test image sets except in frog and mirror because VSVS selects only the two nearest reference views among all pictures in each set. PreSim selects many reference pictures up to 10, excluding the inappropriate reference views. Thus, the time for PreSim is longer than for VSVS. In Tables IX and X, the no-selector algorithm requires the longest time because it uses all reference pictures to synthesize a virtual view. Although the proposed algorithm requires the second-shortest time for most data sets, they are slightly longer than for VSVS. These results imply that the proposed algorithm is very simple compared to conventional methods.

We checked the number of the reference views used to synthesize a virtual view for no-selector, VSVS, PreSim, and the proposed algorithm in Tables XI and XII. As observed in these tables, no-selector uses all reference pictures in each set. The VSVS selects the two nearest reference pictures. In the PreSim algorithm, multiple reference pictures were selected up to 10. Compared with other methods, the proposed algorithm uses fewer reference views, between 3 and 4.

The results in Tables IX, X, XI, and XII indicate that the proposed algorithm is one of the simplest techniques. It selects the most required reference pictures based on minimizing the cost function of (23). In addition, using the selected reference views decreases the algorithm complexity.

VI. CONCLUSION

The performance of a virtual view synthesizer predominantly depends on the set of reference views. This study analyzed the relationship between the number of reference images and the synthesized virtual view quality. Based on the analytical information, we proposed a systematic algorithm to compose the optimal set of reference views. The proposed algorithm consisted of screening and composing steps for the reference pictures. In the screening step, the area of the overlap region, disocclusion, and redundancy were defined with forms of geometric metrics, which were used to exclude some reference pictures that were ineffective in constructing the virtual view. After the screening step, the optimal subset of the remained reference pictures was composed by the iterative method, where the number of reference pictures in the considered subset increased as the iteration was repeated. While composing the subset, hole area, the complementary relationship of the reference pictures, and uniformity of positions of reference pictures were considered based on the geometric measures. One of the challenges is to choose some pictures among the given reference pictures. This work resolved this problem using geometric measures and a systematic procedure. The simulation results demonstrate that the proposed algorithm effectively reduces the hole size and blurring artifacts compared with conventional algorithms.

REFERENCES

[1] Mary-Luc Champel, Igor D.D. Curcio, and Yeshwant Muthusamy, ISO/IEC JTC 1/SC 29/WG 2, N00202 MPEG-I Phase 2 requirements, Online, Apr. 2022.
[2] G. Lafruit, D. Bonatto, C. Tulvan, M. Preda, and L. Yu, “Understanding MPEG-I coding standardization in immersive VR/AR applications,” SMPTE Motion Imaging J., vol. 128, no. 10, pp. 33–39, Nov.—Dec. 2019.
[3] Ralf Schaefer, WG7 members, ISO/IEC JTC 1/SC 29/WG 7, CIP for Dynamic Mesh Coding, Online, Oct. 2021.
[4] J. M. Boyece, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, B. Salahiieh, V. K. M. Vadakital, and L. Yu, “MPEG immersive video coding standard,” in Proc. IEEE, vol. 109, no. 9, pp. 1521–1536, Sep. 2021.
[5] WG 07 MPEG 3D Graphics coding, ISO/IEC JTC 1/SC 29/WG 7 N00267, V-PCC Test Model v17, Online, Jan. 2022.
[6] WG 07 MPEG 3D Graphics coding, ISO/IEC JTC 1/SC 29/WG 7 N00094 G-PCC test model v14 user manual, Online, Apr. 2021.
[7] WG7, MPEG 3D Graphics Coding, ISO/IEC JTC 1/SC 29/WG 7 N283, Perform Analysis of Currently AI-based Available Solutions for PCC, Online, Oct. 2021.
[8] WG7, MPEG 3D Graphics Coding, ISO/IEC JTC 1/SC 29/WG 7 N00289, Lidar EM v4.0: User manual of low-latency and low-complexity codec v4.0 for LiDAR point cloud coding Online, Jan. 2022.
[9] Vinod MV, ISO/IEC JTC 1/SC 29/WG 2 N0157, Use cases and requirements for MIV – edition-2 (final), Online, Jan. 2022.
[10] Xin Jin, Kedeng Tong, Euee S. Jung, Toshiaki Fujii, Gauthier Lafruit, and Mehrdad Teratani, ISO/IEC JTC 1/SC 29/WG 2, N167, Draft Use Cases and Requirements of Lenslet Video Coding (LVC) for Dense Light Fields (DLF), Online, Jan. 2022.
[11] Xin Jin, and Mehrdad Teratani, ISO/IEC JTC 1/SC 29/WG 04, N184, Exploration Experiments and Common Test Conditions for Dense Light Fields, Online, Jan. 2022.
[12] S. Gu, W. Zhang, and R. Wang, "Enhanced DIBR Framework for Free Viewpoint Video," 2021 7th International Conference on Computer and Communications (ICCC), pp. 911–916, 2021.
[13] V. Jantet, C. Guilleminot, and L. Morin, "Object-based layered depth images for improved virtual view synthesis in rate-constrained context," 18th IEEE Int. Conf. Image Processing, pp. 125–128, 2011.
[14] M. Solh and G. AlRegib, "Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video," IEEE J. Selected Topics Signal Process., vol. 6, no. 5, pp. 495–504, Sep. 2012.
[15] I. Ahn and C. Kim, “A novel depth-based virtual view synthesis method for free viewpoint video,” IEEE Trans. Broadcast., vol. 59, no. 4, pp. 614–626, Dec. 2013.
[16] G. Luo, Y. Zhu, Z. Li, and L. Zhang, “A hole filling approach based on background reconstruction for view synthesis in 3D video,” IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 1781–1789, 2016.
[17] C. Lipski, F. Klose, and M. Magnor, "Correspondence and depth-image based rendering a hybrid approach for free-viewpoint video," IEEE Trans. Circuits Systems Video Technol., vol. 24, no. 6, pp. 942–951, Jun. 2014.
[18] M. Tanimoto, T. Fujii, N. Suzuki, and Y. Mori, ISO/IEC JTC1/SC29/WG11 M15377 Reference softwares for depth estimation and view synthesis, 2008.
[19] C. Zhu and S. Li, “Depth image based view synthesis: New insights
and perspectives on hole generation and filling,” IEEE Trans. Broadcast., vol. 62, no. 1, pp. 82–93, Mar. 2016.
[20] S. Li, C. Zhu, and M. Sun, “Hole filling with multiple reference views in DIBR view synthesis,” IEEE Trans. Multimedia, vol. 20, no. 8, pp. 1948–1959, Aug. 2018.
[21] J. Jung and P. Boissonade, ISO/IEC JTC1/SC29/WG11, Versatile View Synthesizer (VVS) 1.0 manual W18076, Macau SAR, Oct. 2018.
[22] S. Fachada, D. Bonatto, A. Schenkel, B. Kroon, and B. Sonneveldt, ISO/IEC JTC1/SC29/WG11, Reference View Synthesizer (RVS) Manual, W18068, Macau SAR, China, Oct. 2018.
[23] B. Salahieh, J. Jung, and A. Dziembowski, ISO/IEC JTC1/SC29/WG04, Test Model 10 for MPEG Immersive Video, N00112, July 2021.
[24] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, and M. Domański, “Multiview synthesis – improved view synthesis for virtual navigation,” 2016 Picture Coding Symp. (PCS), Dec. 2016.
[25] G. Kim, J. Kim, and D. Lee, “Computational complexity of view synthesis with the number of selected images using array cameras,” 2020 IEEE Int. Conf. Consumer Electronics–Asia (ICCE-Asia), pp. 1–3, 2020.
[26] A. Vetro, W. Matusik, H. Pfister, and J. Xin, “Coding approaches for end-to-end 3D TV systems,” Picture Coding Symposium 2004 (PCS 2004), San Francisco, California, USA, Dec. 2004.
[27] S. Shimizu, G. Bang, D. B. Graziosi, T. Senoh, M. P. Tehranl, A. Vetro, K. Wegner, G. Lafruit, and M. Tamimoto, ISO/IEC JTC1/SC29/WG11, Use cases and requirements on free-viewpoint television (FTV), N16130, San Diego, USA, Feb. 2016.
[28] Adrian Dziembowski, Dawid Mieloch, and Jaroslaw Samelak, ISO/IEC JTC 1/SC 29/WG 04 m57753, 3D-HEVC in MIV verification tests, Online, Oct. 2021.
[29] T. Maugey, G. Petrazzoli, P. Frossard, M. Cagnazzo, and B. Pesquet-Popescu, “Reference view selection in DIBR-based multiview coding,” IEEE Trans. Image Process., vol. 25, no. 4, pp. 1808–1819, Apr. 2016.
[30] T. Maugey, G. Petrazzoli, P. Frossard, M. Cagnazzo, and B. Pesquet-Popescu, “Key view selection in distributed multiview coding,” IEEE Visual Commun. and Image Processing Conf., pp. 486–489, 2014.
[31] A. Dziembowski, J. Samelak, and M. Domański, “View selection for virtual view synthesis in free navigation systems,” Int. Conf. Signals and Electronic Systems (ICSES), pp. 83–87, 2018.
[32] H. Yuan and R. C. Veltkamp, “PreSim: A 3D photo-realistic environment simulator for visual AI,” IEEE Robotics and Automation. Lett., vol. 6, no. 2, pp. 2501–2508, Apr. 2021.
[33] D.-Y. Nam, H.-K. Kim, and J.-K. Han, “Efficient view synthesis algorithm using view selection for generating 6DoF images,” IEEE 23rd Int. Workshop Multimedia Signal Processing (MMSP), Tampere, Finland, 6–8 Oct. 2021.
[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Greedy algorithms,” Introduction to Algorithms, MIT Press, ISBN 978-0-262-03293-3, pp. 414–450, 2001.

Da-Yoon Nam was born in Chuncheon, Korea, in 1997. Currently, she is pursuing her MS in electrical engineering at Sejong University, Seoul, Korea. Her research interests include 360 virtual reality, high efficiency video coding, and versatile video coding (VVC). She has been participating in the standardization of VVC since 2019.

Woo-Kyung Jung was born in Seoul, Korea, in 1995. He received his BS degree in electrical engineering from Sejong University, Seoul, Korea. He is pursuing his MS degree in Sejong University. His research interests include computer vision, 360 VR, high efficiency video coding (HEVC), versatile video coding (VVC), and Deep-learning based image processing.

Hyeon-Deok Han was born in Daejeon, Korea in 1996. He received his BS degree in electrical engineering from Sejong University, Seoul, Korea. He is pursuing his MS degree in Sejong University. His research topics are mainly computer vision and 360 VR. In addition, he is proficient in the field of video coding based on high efficiency video coding (HEVC) and versatile video coding (VVC).

Jong-Ki Han was born in Seoul, Korea, in 1968. He received his BS, MS, and PhD in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1992, 1994, and 1999, respectively. From 1999 to 2001, he was a member of the technical staff in the corporate R&D center at Samsung Electronics Company in Suwon, Korea. He is currently a professor in the Department of Electrical Engineering, Sejong University, Seoul, Korea. His research interests include image and video coding using high efficiency video coding and versatile video coding (VVC) and image coding and processing for 360 virtual reality. He participated in the standardization of HEVC and has been participating in the VVC standard since 2016.