Quality Improvement for Real-time Free Viewpoint Video Using View-dependent Shape Refinement

Ryosuke Watanabe (member)†, Tomoaki Konno †, Keisuke Nonaka (member)†, Tatsuya Kobayashi (member)†, Haruhisa Kato (member)†, Masaru Sugano (member)†

Abstract In this paper, we propose a quality improvement method of a real-time free viewpoint video (FVV) synthesis. Two functions, adaptive silhouette dilation and view-dependent shape refinement, are applied to obtain a 3D model with accurate shapes in the real-time FVV. The adaptive dilation reduces the missing part of a reconstructed 3D model caused by a camera calibration error. In addition, the excessively expanded part of a 3D model caused by the silhouette dilation is corrected by the view-dependent refinement algorithm on a screen rendering. A part of the model outlines becomes transparently displayed to refine the model shape by comparing the camera image used for texture mapping with a background model. In the experiments, the proposed method achieved the best quality compared with conventional real-time FVV methods. Furthermore, the processing was executed in real-time because the computation cost of the proposed functions was extremely small.

Key words: free viewpoint video, 3D reconstruction, real-time processing, 3D computer vision, camera calibration, image processing

1. Introduction

Free viewpoint video (FVV) technology has been the focus of much attention in the computer vision field. The FVV enables users to watch a 3D scene from a viewpoint where real cameras have not been mounted. Moreover, a walk-through experience, such as walking around in a 3D space, can be realized by wearing an XR device (e.g. VR headsets, AR glasses, and the like). In particular, the FVV is expected to be used for various sports applications such as watching a sports game for entertainment and analyzing the motions of athletes to improve their performance.

Many approaches for FVV generation have been proposed so far. Among them, a model-based method which reconstructs a 3D computer graphics model from multiple cameras has been intensively studied. Considering the use for sports watching, the model-based method is suitable because the user can select arbitrary viewpoints compared with other types of FVV, for example, a camera switching-based FVV and light field-based FVV. One of the most well-known model-based approaches is visual hull reconstruction. As the first step for the visual hull reconstruction, a silhouette image of an object is extracted for all RGB cameras. After that, a 3D computer graphics model is reconstructed as the intersection of the visual cones from the silhouette images as shown in Fig. 1(a). However, it takes enormous time to reconstruct a 3D model in such a space as a large sports field. Thus, a real-time feature is necessary for achieving the live sports watching using FVV. To solve the problem, a few model-based FVV generation methods based on visual hull having a real-time feature are proposed in recent years. A coarse-to-fine voxelization method and a multiple planes-based method accelerate the calculation process of visual hull reconstruction. These two methods have achieved generation of a 3D model in real-time with multiple RGB cameras in a stadium.

However, the visual quality of the conventional real-time FVV methods still has problems. First, it is difficult to accurately extract silhouette images depending on shooting environments. For example, sensor noises and lighting changes such as shadows are extracted incorrectly as a foreground. Considering FVV production for a stadium-scale sports, the angle of view of real cameras must be set to capture a wide area. At this time, multiple images with high-resolution are required because the resolution of an object such as a player becomes relatively low. Although accurate silhouette
Fig. 1 Visual hull reconstruction from silhouette images: (a) ordinary visual hull reconstruction, (b) visual hull reconstruction including errors caused by estimated camera position, (c) visual hull reconstruction with dilated silhouette images.

The remainder of this paper is organized as follows: Section 2 describes conventional related work. Section 3 explains the proposed real-time FVV generation method using the adaptive silhouette dilation and the view-dependent shape refinement. Section 4 describes the experiments for evaluating the proposed method by a comparison with some conventional real-time FVV technologies. Finally, we conclude the paper in Section 5.

2. Related work

2.1 Overview of model-based free viewpoint video

A model-based FVV is classified into a billboard-based method\cite{10,11,12,13} and a volumetric model-based method\cite{6,7,8,9,14,15,16,17,18,19,20}. The billboard-based method builds a 2D plane called a "billboard" for each object in a 3D space. After that, the texture from the nearest camera is projected onto the billboard. According to the previous study\cite{27}, the billboard-based method is robust to a calibration noise compared with the volumetric model-based method. Also, several real-time calculation methods based on the billboard concept have been proposed\cite{12,13}. The billboard model, however, sometimes looks unnatural because the shape of the 3D object approximates a 2D plate. For example, when an object is observed from directly above, it just looks like a thin plane because the billboard stands vertically to the ground.

In contrast, a method using a volumetric 3D model focuses on accurate reconstruction of the 3D shape. Recently, a method based on deep-learning has achieved a
high reconstruction accuracy. However, the methods require many training datasets and enormous calculation time. Also, a method using an RGB-D sensor has been proposed to reconstruct a volumetric 3D model. However, when the distance between a sensor and a captured object is long, depth information cannot be acquired because of the limitation of a depth sensor’s functionality. Furthermore, a method based on visual hull reconstruction has been studied intensively. Since the visual hull concept was proposed in the original paper, many studies have improved the 3D reconstruction quality and the processing time of the visual hull reconstruction. The concept of a conservative visual hull improves the reconstruction accuracy of the 3D model shapes. Also, a method using photo-consistency has been proposed to improve the image quality of the visual hull-based FVV. However, these methods cannot achieve the real-time processing in a large space such as a stadium. To reduce the calculation time, coarse-to-fine visual hull generation is proposed for a sports scene. Maeda et al. have proposed the method using coarse-to-fine visual hull generation and micro facet billboard. In this method, a voxel model generated by the corase-to-fine voxelization is converted to a small billboard for displaying in real-time. Therefore, the problem regarding the unnatural shape which is a drawback of the billboard-based method still remains. After that, Chen et al. have succeeded in generating a volumetric polygon model by using the coarse-to-fine visual hull generation. Although this method succeeds in generating a 3D model in real-time, the silhouette extraction process before visual hull reconstruction is not in real-time because an instance segmentation method having a heavy processing time is adopted to improve the silhouette extraction quality. Also, multiple planes-based methods calculate the 3D shape of an object by integrating a set of multiple planes in real-time. First, Matsuyama et al. achieve real-time FVV generation using multiple planes-based concept with a computer cluster. After that, Nonaka et al. have extended this concept to improve the quality of FVV by using view-dependent multiple planes. These methods achieve real-time processing including the silhouette extraction process. However, the quality and the computation cost depend on the number of planes because the 3D shape is approximated to a set of planes. In the conventional real-time FVV generation using a volumetric model, the real-time processing was only confirmed in a relatively narrow space such as a batter’s box only in a baseball stadium and a volleyball court. If a large 3D scene such as a soccer stadium is targeted, a reconstructed 3D model becomes coarse to maintain the real-time processing.

2.2 Camera calibration technology

For 3D reconstruction in a model-based FFV, camera calibration is necessary to associate 3D world coordinates with image coordinates. The reconstruction quality depends on the estimated internal and external camera parameters as shown in Fig. 1(b). Although many studies use a calibration object such as a checkerboard to estimate the parameters, it is difficult to set it in a sports stadium during a game. In addition, in the case that re-calibration is required due to an accidental touch of the camera position, the calibration object approach is difficult to adopt for FVV production in a stadium.

Under such a situation, a method using a priori information such as intersections of field lines has been proposed for FVV technology. In the method, the field line’s intersections detected in a 2D image are associated with those in 3D world coordinates. To reduce the calibration errors, the methods focus on improving the accuracy of the intersection detection. Yao et al. classify detected field lines into horizontal and vertical directions to find intersections. In addition, the flexible intersection selection method improves the accuracy of the field line-based method. However, these methods require many field lines to estimate accurate camera parameters. Therefore, the applicable sports and camera angles are limited. In addition, a field area is detected by using the color threshold decided by a human in advance. Generally, the color of the field is different according to the stadium and lighting condition. This process implies that parameter tuning by trial and error is required to obtain a high accuracy.

Also, to improve the calibration accuracy of sparsely deployed multi-view fixed cameras, the method using a moving camera has been proposed. This approach is effective in the scene where a camera can be moved in a sports stadium. However, in some practical cases, it is difficult to move a camera due to restrictions of the setting cost and the range where the camera can be moved with permission. Therefore, in this paper, we assume that all cameras are fixed in a stadium.

2.3 Silhouette extraction technology

Silhouette extraction is one of the most important elemental technologies in FVV because the quality of
a reconstructed 3D model based on the visual hull depends on the silhouette shape.

In the visual hull reconstruction, many silhouettes are required to obtain an accurate visual hull because the silhouettes from all cameras affect the shape of the 3D model. Thus, for real-time FVV generation, silhouette extraction must be processed in real-time for multiple camera images. In particular, when we capture a large space such as a soccer stadium, a high-resolution image is required for the acquisition of a clear object texture. However, there are few silhouette extraction methods that tackle real-time processing for multiple high-resolution images. Generally, a state-of-the-art method does not focus on the real-time processing for multiple cameras\(^{24} - 26\).

In a conventional real-time FVV method using multiple planes\(^{27}\), an object silhouette is extracted by a background subtraction method using single Gaussian distribution\(^{24}\). The method succeeds in extracting a silhouette for 16 Full-HD images in real-time.

### 3. Proposed method

Figure 2 shows the production process of the proposed method. In the system, camera calibration is executed before a game starts, as described in subsection 3.1. After a game starts, the three processes, the silhouette extraction (subsections 3.2 & 3.3), 3D model reconstruction (subsection 3.4), and rendering (subsection 3.5), are performed for all frames in real-time. In this paper, the processing faster than 30 fps is defined as the real-time processing. We assumed the FVV calculation system in which the three processes are assigned to the three computers to maintain the real-time. If each process can be performed faster than 30 fps, all the processes can be performed in real-time by pipeline processing.

#### 3.1 Camera calibration

First, we perform camera parameter estimation after fixing an angle of all \(C\) cameras. We prepare a set of feature point positions \(G_{\text{ALL}} = \{g_0(x, y, z), g_1(x, y, z), \ldots, g_{J-1}(x, y, z)\}\) such as the field line’s intersections in 3D stadium coordinates in advance, as shown in Fig. 3(a). The number of all intersections is denoted by \(J\). The positions are prepared from the dimensional standard of field lines. After that, the calibration process is individually processed for each camera before a game starts as shown in Fig. 3. For simplification, we explain the \(c\)-th camera as follows. A set of feature points \(F_c = \{f_0(u, v), f_1(u, v), \ldots, f_{K-1}(u, v)\}\) is manually clicked on a screen for all cameras, where \(K\) indicates the number of all visible feature points in the image. After obtaining the set of feature points \(F_c\) by a human operator, internal and external camera parameters are estimated by using the conventional camera calibration method\(^{28}\) using field line intersections. A set of the feature points in 3D stadium coordinates \(G_c = \{g'_0(x, y, z), g'_1(x, y, z), \ldots, g'_{K-1}(x, y, z)\}\) is correlated with \(F_c\) to estimate the camera parameters.

As described in subsection 2.2, the method\(^{29}\) detects an intersection automatically. However, parameter tuning is required, and the applicable scene is limited. As long as a camera is fixed, a human operator manually obtains a set of feature points \(F_c\) only once. Therefore, the operation is rational and practical in this use case.

After the camera calibration, \(G_c\) is projected onto a camera plane by the estimated camera parameters. A set of the projected points is defined as \(H_c = \{h_0(u, v), h_1(u, v), \ldots, h_{K-1}(u, v)\}\). The set of input feature points \(F_c\) is located near the projected feature points \(H_c\) if we succeed in the camera calibration. In the proposed system, the set of projected points \(H_c\) is used to determine the amount of the silhouette dilation. The details are described in subsection 3.3.

#### 3.2 Silhouette extraction

Next, an object silhouette is extracted by a conventional background subtraction method\(^{24}\) using a single Gaussian. As with the previous real-time FVV study\(^{27}\), a background model is constructed as single Gaussian distribution which is composed of the mean \(\mu_c(u, v)\) and the standard deviation \(\sigma_c(u, v)\). The mean and the standard deviation are calculated by the same process as the conventional method\(^{24}\). After acquiring Gaussian distribution, a silhouette is extracted by the background subtraction method between a camera image and the background model constructed by the single Gaussian distribution.

Since this method is suitable for parallel processing for each pixel, we succeeded in real-time processing for 20 Full-HD cameras in the experiments.

#### 3.3 Directional and non-directional silhouette dilation

After extracting a silhouette by using the conventional background subtraction, we perform directional and non-directional silhouette dilation for reducing the missing part of a 3D model caused by calibration errors. As shown in Fig. 1(b), the 3D model calculated by vi-
Fig. 2  FVV Generation process of the proposed system.

Fig. 3  Camera calibration: (a) a priori information for camera calibration, (b) camera calibration and error estimation process.

Fig. 4  Directional and non-directional silhouette dilation.

Fig. 5  Relationship between the directional and non-directional silhouette dilation in the soccer scene.

Figure 4 explains the directional and non-directional silhouette dilation. Here, we consider estimating the calibration errors of a pixel \((u, v)\) in the \(c\)-th camera image. In the proposed method, both directional and non-directional dilation are executed for each pixel as shown in Fig. 4.

First, we describe the details of the directional silhouette dilation. Near one of the input feature points \(f_k(u, v)\) on a 2D image, the camera calibration errors are approximately estimated as the difference between \(f_k(u, v)\) and the projected points \(h_k(u, v)\), where \(k\) in-
icates the index of the feature points. Here, the error vector $e_k = (e_u, e_v)$ is calculated as
$$e_k = h_k - f_k.$$ (1)

When the silhouette is dilated in response to the error vector, the missing part is restored. In the directional dilation, therefore, a dilated vector $d_{DIR} = (d_u, d_v)$ for the directional dilation is calculated using the error vector $e_n$ as denoted by
$$d_{DIR} = (D_{MAX} - D_n) e_n.$$ (2)

Here, $n$ shows the index of the nearest feature point from the pixel $(u, v)$. $D_n$ represents the L2 distance from the pixel $(u, v)$ to the nearest feature point $f_n$. Also, $D_{MAX}$ is equal to the maximum $D_n$ when $D_n$ is calculated for all pixels.

Second, we describe the non-directional dilation. In the conventional calibration method, camera parameters are optimized by associating the feature points $F_c$ on the 2D image with a priori 3D positions $G_c$ such as the field line’s intersections. However, when the distance between a pixel $(u, v)$ and the nearest feature point $f_n$ is long, it is difficult to estimate the error vector $e_k$ calculated by Eq. (1) because there are no associated points that can be used as clues. Although the direction estimation is difficult, the magnitude of the errors becomes large according to the distance from the nearest feature point. Therefore, the amount of non-directional silhouette dilation is changed in response to the distance from the nearest feature point $f_n$ to the pixel $(u, v)$. For example, at the lower right pixel in Fig. 5, non-directional dilation is mainly used to prevent causing the missing part on a 3D model because the distance between the pixel and the nearest feature point is extremely long. The filter size for the non-directional dilation $d_{ND}(u, v)$ is calculated by
$$d_{ND} = \frac{2D_n}{D_{MAX}} D_{AVE},$$ (3)

where the average L2 distance $D_{AVE}$ is shown as
$$D_{AVE} = \frac{1}{K} \sum_{k=0}^{K-1} \| H_k - F_k \|^2.$$ (4)

According to Eq. (3), $d_{ND}$ increases in proportion to the distance $D_n$ between the pixel $(u, v)$ and the nearest feature point $f_n$.

Over all pixels $(u, v)$, the $d_{DIR}$ and $d_{ND}$ are independently calculated. Also, decimals of $d_{DIR}$ and $d_{ND}$ are rounded to the nearest whole number. In the processing, $d_{DIR}$ and $d_{ND}$ can be calculated in advance because the two parameters are not changed frame-by-frame because the parameters depend on the calibration errors. Therefore, the increased calculation time of the directional and non-directional dilation becomes extremely small.

3.4 3D shape reconstruction using coarse-to-fine voxelization

In the 3D shape reconstruction process, a conventional method using a coarse-to-fine voxelization strategy is applied. As described in subsection 2.1, this method achieves real-time processing in the reconstruction process. As with the original paper, an output voxel model is converted to a polygon model by using the marching cube method.

3.5 Rendering with view-dependent shape refinement

After generating 3D polygon models, a 2D image viewed from a selected viewpoint $r_{view}$ is rendered. The rendering process employs the view-dependent texture mapping introduced in the previous study. As the difference between the proposed method and the previous study, a part of the reconstructed 3D model on the rendered image becomes transparent in the texture mapping process. The detailed rendering process is described as follows.

1) View-dependent texture mapping

After determining the position of a selected viewpoint, 3D models are colored by texture mapping. Ba-
sically, a reference camera used for texture mapping is decided as the nearest camera by calculating the angle distance from $r_{view}$ to each camera. If a polygon cannot be seen from a real camera because of the occlusion with other polygons, the camera is not used for texture mapping to the polygon. As a result, the nearest non-occluded camera is adopted for texture mapping.

(2) View-dependent model shape refinement

In the proposed method, some of the pixels $(\xi, \eta)$ in a synthesized image become transparent when each polygon $P_b (b \in \{1, \ldots, B\})$ is drawn, where $B$ indicates the number of polygons. When the texture mapping is performed for a polygon, we calculate the difference between the camera image for the mapping and the background model updated by the silhouette extraction. If the difference is small, the $(\xi, \eta)$ becomes transparent to refine the excessively expanded model shape caused by the silhouette dilation, as shown in Fig. 6. We describe the details of determining transparency when the polygon $P_b$ is drawn to the pixel $(\xi, \eta)$. Here, an example polygon $P_1$ is considered as shown in Fig. 7. The polygon $P_1$ is composed of three vertices $V_1 (x_{v1}, y_{v1}, z_{v1})$, $V_2 (x_{v2}, y_{v2}, z_{v2})$, and $V_3 (x_{v3}, y_{v3}, z_{v3})$. $(\xi_1, \eta_1)$, $(\xi_2, \eta_2)$, and $(\xi_3, \eta_3)$ are the pixels on the rendered image plane when $V_1$, $V_2$, and $V_3$ are projected onto the rendered image plane, respectively. Also, $(u_1, v_1)$, $(u_2, v_2)$, and $(u_3, v_3)$ are the pixels on the real camera plane for texture mapping when $V_1$, $V_2$, and $V_3$ are projected onto the rendered image plane, respectively. In this case, the pixel position $(u, v)$ on the real camera is obtained by linear interpolation based on the positional relationship between the pixel $(u, v)$ and the vertices constructing a polygon $(u_1, v_1)$, $(u_2, v_2)$, and $(u_3, v_3)$. After that, the following equation is calculated.

$$|I_m(u, v) - \mu_m(u, v)| < T_R,$$

where $I_m(u, v)$ indicates the intensity of a camera image for texture mapping. $m$ represents the selected camera index for texture mapping. Also, $T_R$ is the user-defined threshold. If Eq. (5) is true, the pixel $(\xi, \eta)$ becomes transparent when the polygon $P_1$ is drawn. In reality, Eq. (5) is calculated for all drawing polygons.

Furthermore, on the rendered image $(\xi, \eta)$, Eq.(5) is calculated only for the pixels near the edge of the object. If the pixel value $I_m(u, v)$ is close to $\mu_m(u, v)$, a region that should not be transparent may be missing. This false decision can be reduced by limiting the calculation area. Specifically, a $(2a + 1) \times (2a + 1)$ pixels area centered on the pixel $(\xi, \eta)$ is defined on a rendered image, where $a$ shows a user-defined filter size for the view-dependent shape refinement. If all the pixels in the area belong to drawing polygons, Eq. (5) is not calculated because the pixel $(u, v)$ does not locate at the outline of the model shape.

4. Experiments

4.1 Experimental conditions

Two datasets including multiple videos were used for evaluation. One is the public dataset\(^{36,37}\) regarding a soccer game captured from 20 RGB cameras. The other is our dataset of a baseball game captured from 16 cameras focusing on the batter’s box. Figure 8 shows the camera arrangement of each sports scene. A 3D model is reconstructed in the area where all cameras commonly shoot. The videos in the dataset were captured at 30 fps, and the resolution is $1920 \times 1080$. Likewise, the screen resolution for rendering was set to $1920 \times 1080$ throughout all experiments. For evaluating the processing time, we used a computer with AMD Ryzen Threadripper 2970 WX CPU, 128.0 GB RAM, and two NVIDIA GeForce RTX 2080 Ti GPUs (with NVLink connection).

In the following experiments, we performed quantitative evaluation to confirm the FVV quality. As an evaluation metric, Precision (Prec), Recall (Rec), and F-Measure (FM) were used. In the evaluation, the rendering was performed at the position of a real camera. After that, we calculated the metrics at all real camera positions between a shape mask and a ground-truth silhouette which was manually prepared by human annotation. The shape mask was output by projecting 3D models onto the pixel on a rendered image plane as shown in Fig. 9. In the shape mask, the pixels where the 3D model is drawn are regarded as a foreground. Note that 3D model outlines which were not displayed by the view-dependent refinement were regarded as a background in the shape mask. In addition, in the experiment 2 as shown in subsection 4.3, we used SSIM (Structural Similarity)\(^{30}\) between a real camera image and a rendered image to clarify the video quality compared with conventional methods.

4.2 Experiment 1 : Quality evaluation of proposed function

We prepared 12 conditions to confirm the effect of the proposed functions, adaptive dilation and 3D shape refinement, as shown in Table 1. Condition #1 in Table 1 indicates a conventional case without using the silhouette dilation and the view-dependent refine-
Converging points of all cameras

Production area 40m(x) x 5m(y) x 30m(z)

Converging points of all cameras

Production area 8m(x) x 3m(y) x 8m(z)

Fig. 8  Real camera arrangements: (a) soccer scene, (b) baseball scene.

Fig. 9  Model shape evaluation using the shape mask.

Table 1  Quality evaluation of the proposed functions using 12 parameter settings. Blue characters show the highest F-Measure in soccer, baseball, and the average.

| Conditions | Silhouette dilation | View-dependent refinement | Prec | Rec | FM  | Prec | Rec | FM  | Average F-Measure |
|------------|---------------------|----------------------------|------|-----|-----|------|-----|-----|------------------|
| #1         | None                | None                       | 0.844| 0.620| 0.714| 0.936| 0.919| 0.927| 0.821            |
| #2         | None                | TR = (20,10,10)            | 0.875| 0.616| 0.722| 0.955| 0.902| 0.927| 0.825            |
| #3         | None                | TR = (30,15,15)            | 0.898| 0.606| 0.722| 0.964| 0.882| 0.921| 0.821            |
| #4         | Constant (d=3)     | None                       | 0.819| 0.857| 0.718| 0.829| 0.984| 0.980| 0.809            |
| #5         | Constant (d=3)     | TR = (20,10,10)            | 0.759| 0.839| 0.796| 0.913| 0.951| 0.931| 0.864            |
| #6         | Constant (d=3)     | TR = (30,15,15)            | 0.810| 0.813| 0.810| 0.931| 0.929| 0.930| 0.870            |
| #7         | Constant (d=6)     | None                       | 0.474| 0.941| 0.630| 0.736| 0.992| 0.845| 0.737            |
| #8         | Constant (d=6)     | TR = (20,10,10)            | 0.702| 0.916| 0.793| 0.883| 0.963| 0.921| 0.857            |
| #9         | Constant (d=6)     | TR = (30,15,15)            | 0.765| 0.882| 0.818| 0.902| 0.949| 0.925| 0.871            |
| #10        | Adaptive            | None                       | 0.590| 0.878| 0.703| 0.849| 0.982| 0.910| 0.807            |
| #11        | Adaptive            | TR = (20,10,10)            | 0.748| 0.859| 0.798| 0.918| 0.950| 0.944| 0.866            |
| #12        | Adaptive            | TR = (30,15,15)            | 0.785| 0.863| 0.821| 0.925| 0.934| 0.930| 0.875            

According to Table 1, when both the silhouette dilation and the view-dependent refinement were used in combination (#5, #6, #8, #9, #11, and #12), better qualities were obtained. If either only one of the two was used, the quality cannot be improved compared with the conventional pattern #1. As the reason for this, the silhouette dilation causes the increase of an excessively expanded part of a 3D model instead of reducing the missing part. In contrast, the view-dependent shape refinement has an adverse trend. However, only when $TR(H,S,V) = (30,15,15)$ were tested to check the effects of the threshold difference. In addition, the filter size for the view-dependent shape refinement $a$ was set to 10 in the experiments. In the original coarse-to-fine voxelization paper, the voxel size was set to 2 cm. However, in the soccer scene, the processing time was not in real-time because the 3D space was very large. Therefore, the voxel size $V_P$ of all conditions #1 – #12 was commonly set to 3 cm to maintain the real-time processing.

Also, the conditions #2 and #3 adopted the view-dependent refinement only. By contrast, in the conditions #4 – #9, all pixels on a silhouette image were constantly dilated by the non-directional dilation using $(2d + 1) \times (2d + 1)$ filter. In contrast, adaptive silhouette expansion using the directional and non-directional dilation was applied to the conditions #10 – #12. Also, Eq. (5) was calculated in the HSV color space in the experiments. When Eq. (5) was true for all color spaces, the pixel was transparently displayed. Two types of threshold $TR(H,S,V) = (20,10,10)$ and $TR(H,S,V) = (30,15,15)$ were tested to check the effects of the threshold difference. In addition, the filter size for the view-dependent shape refinement $a$ was set to 10 in the experiments. In the original coarse-to-fine voxelization paper, the voxel size was set to 2 cm. However, in the soccer scene, the processing time was not in real-time because the 3D space was very large. Therefore, the voxel size $V_P$ of all conditions #1 – #12 was commonly set to 3 cm to maintain the real-time processing.
we combine the two functions, the expanded and the missing part can be reduced efficiently.

Also, we discuss the effect of the adaptive dilation. When we compared under the same $T_R$ conditions, the condition #11 was better than the conditions #5 and #8. Likewise, the F-Measure of condition #12 was better than those of conditions #6 and #9, too. When we paid attention to the difference between the scenes, the conditions with $d = 3$ was better than those with $d = 6$ in the baseball scene, but the opposite trend was seen in the soccer. These results imply that the optimal amount of silhouette dilation is dependent on the scene. In the adaptive dilation, the F-Measure was higher both the baseball and the soccer scene because the different amount of silhouette dilation was set to each scene considering calibration errors. In addition, Fig. 10 shows the effects of the directional dilation. The missing part can be reduced in the red rectangle in Fig. 10(c) compared with Fig. 10(a) and Fig. 10(b). Furthermore, the excessively expansion parts of the condition #12 was much smaller than those of the condition #9 as shown in the blue and yellow rectangles in Fig. 10. Therefore, we succeeded in improving the quality while suppressing the unnecessary artifacts by using the directional dilation. Thus, the conditions with adaptive dilation were better than those with the constant dilation because the optimal amount of silhouette dilation was estimated automatically.

In addition, while #11 achieved the highest F-Measure in the baseball scene, #12 was the best in the soccer scene. In the experiment, the calibration errors in the soccer scene were larger than those in the baseball scene because $D_{AVG}$ of the soccer was twice bigger than that of the baseball. As a result, the silhouette images of the soccer scene were dilated many times. Therefore, the high-threshold is suitable for the soccer scene to suppress the expanded parts on a 3D model.

4.3 Experiment 2: Comparison with conventional methods

Two conventional real-time FVV methods, the coarse-to-fine voxelization method (C2F)\textsuperscript{7} and the fast plane method (FastP)\textsuperscript{9}, were compared with the proposed method in terms of the quality and the processing time.

In the experiment, the conventional methods used the same camera parameters as the proposed method. Regarding the C2F\textsuperscript{7}, the fine voxel grid size $V_F$ was set to 2 cm which is the same parameter as the original paper\textsuperscript{7}, or 3 cm which is the same as the proposed method as discussed in subsection 4.2. Also, the computation cost of the FastP\textsuperscript{9} depends on the number of planes $P$ and the size of the production area for FVV generation, as described in subsection 2.1. Although the number of planes was set to 800 as in the original paper\textsuperscript{9}, the real-time processing was not possible in the soccer scene because the production area was large. According to our preliminary experiments, the maximum number of planes to be able to maintain the processing speed at faster than 30 fps was 170. Therefore, we selected $P = 800$ and $P = 170$ in this experiment.

(1) Experiment 2-1: Reconstruction quality comparison

This subsection shows the results of quantitative evaluation of the quality. Like the experiment described in subsection 4.2, a shape mask was compared between the proposed method and the conventional real-time FVV methods\textsuperscript{7,9} as shown in Table 2 and Table 3. In addition, we calculated SSIM\textsuperscript{38} between a real camera image and a rendered image generated by the proposed method and conventional real-time FVV methods\textsuperscript{7,9} to compare the video quality as shown in Table 4 and Table 5. Regarding the evaluation with SSIM, two kinds
Table 2  Quantitative evaluation at real camera position of the proposed method and conventional real-time FVV methods, C2F\(^7\) and FastP\(^9\). Blue characters show the highest F-Measure in soccer, baseball, and the average.

| Method      | Soccer | Baseball | Average | F-Measure |
|-------------|--------|----------|---------|-----------|
|             | Prec   | Rec      | FM      | Prec      | Rec    | FM      | Prec | Rec    | FM      | Prec  | Rec    | FM      | F-Measure |
| C2F\(^7\) (V = 2cm) | 0.729 | 0.803 | 0.763 | 0.908 | 0.963 | 0.934 | 0.849 |
| C2F\(^7\) (V = 3cm) | 0.730 | 0.800 | 0.761 | 0.904 | 0.969 | 0.931 | 0.846 |
| FastP\(^9\) (P = 800) | 0.760 | 0.738 | 0.738 | 0.926 | 0.944 | 0.935 | 0.857 |
| FastP\(^9\) (P = 170) | 0.775 | 0.645 | 0.702 | 0.934 | 0.932 | 0.935 | 0.817 |
| Proposed (#11) | 0.748 | 0.859 | 0.798 | 0.918 | 0.950 | 0.934 | 0.866 |
| Proposed (#12) | 0.785 | 0.863 | 0.821 | 0.925 | 0.934 | 0.930 | 0.875 |

Table 3  Quantitative evaluation at a virtual viewpoint of the proposed method and conventional real-time FVV methods, C2F\(^7\) and FastP\(^9\). Blue characters show the highest F-Measure in soccer, baseball, and the average.

| Method      | Soccer | Baseball | Average | F-Measure |
|-------------|--------|----------|---------|-----------|
|             | Prec   | Rec      | FM      | Prec | Rec | FM | Prec | Rec | FM | F-Measure |
| C2F\(^7\) (V = 2cm) | 0.709 | 0.805 | 0.754 | 0.876 | 0.964 | 0.918 | 0.836 |
| C2F\(^7\) (V = 3cm) | 0.710 | 0.802 | 0.753 | 0.875 | 0.962 | 0.917 | 0.835 |
| FastP\(^9\) (P = 800) | 0.722 | 0.737 | 0.728 | 0.899 | 0.947 | 0.922 | 0.824 |
| FastP\(^9\) (P = 170) | 0.754 | 0.650 | 0.696 | 0.908 | 0.932 | 0.919 | 0.808 |
| Proposed (#11) | 0.750 | 0.854 | 0.781 | 0.894 | 0.950 | 0.925 | 0.857 |
| Proposed (#12) | 0.761 | 0.858 | 0.807 | 0.897 | 0.939 | 0.917 | 0.862 |

Table 4  Quantitative evaluation using SSIM at real camera position of the proposed method and conventional real-time FVV methods, C2F\(^7\) and FastP\(^9\). Blue characters show the highest SSIM in soccer, baseball, and the average.

| Method      | SSIM (All pixels) | SSIM (Around objects) |
|-------------|-------------------|-----------------------|
|             | Soccer | Baseball | Average | Soccer | Baseball | Average |
| C2F\(^7\) (V = 2cm) | 0.498 | 0.668 | 0.583 | 0.551 | 0.793 | 0.672 |
| C2F\(^7\) (V = 3cm) | 0.498 | 0.668 | 0.583 | 0.549 | 0.792 | 0.671 |
| FastP\(^9\) (P = 800) | 0.493 | 0.668 | 0.584 | 0.549 | 0.790 | 0.580 |
| FastP\(^9\) (P = 170) | 0.492 | 0.667 | 0.580 | 0.536 | 0.783 | 0.570 |
| Proposed (#11) | 0.501 | 0.659 | 0.585 | 0.600 | 0.793 | 0.697 |
| Proposed (#12) | 0.500 | 0.667 | 0.584 | 0.593 | 0.780 | 0.687 |

Table 5  Quantitative evaluation using SSIM at virtual viewpoint of the proposed method and conventional real-time FVV methods, C2F\(^7\) and FastP\(^9\). Blue characters show the highest SSIM in soccer, baseball, and the average.

| Method      | SSIM (All pixels) | SSIM (Around objects) |
|-------------|-------------------|-----------------------|
|             | Soccer | Baseball | Average | Soccer | Baseball | Average |
| C2F\(^7\) (V = 2cm) | 0.494 | 0.660 | 0.577 | 0.500 | 0.727 | 0.614 |
| C2F\(^7\) (V = 3cm) | 0.493 | 0.659 | 0.576 | 0.499 | 0.725 | 0.612 |
| FastP\(^9\) (P = 800) | 0.488 | 0.659 | 0.574 | 0.324 | 0.723 | 0.524 |
| FastP\(^9\) (P = 170) | 0.486 | 0.658 | 0.572 | 0.317 | 0.715 | 0.516 |
| Proposed (#11) | 0.497 | 0.660 | 0.579 | 0.547 | 0.730 | 0.639 |
| Proposed (#12) | 0.496 | 0.658 | 0.577 | 0.539 | 0.716 | 0.628 |

of regions we named as "All pixels" and "Around objects" were evaluated as shown in Table 4 and Table 5. "All pixels" indicates that all the pixels of a camera image was evaluated for acquiring SSIM results. However, it was difficult to know the difference of evaluation values because the region where there was a 3D reconstructed model on the image was extremely narrow. Therefore, in the "Around objects" pattern, we evaluated in the limited area inside the bounding box which circumscribes to the foreground outline of the ground-truth silhouette.

In Table 2 and Table 4, all cameras were used for 3D reconstruction. In contrast, in Table 3 and Table 5, one camera was not used for 3D reconstruction. In other words, 19 cameras and 15 cameras were used for 3D reconstruction in the soccer and the baseball, respectively. The F-Measure and SSIM were calculated from the position of the unused camera. By repeating this process for all cameras, the average F-Measure and SSIM were calculated. Therefore, Table 3 and Table 5 show the quality from a viewpoint where a real camera was not mounted.

From the result of Table 2 and Table 3, the proposed method achieved the highest average F-Measure. Also, Table 4 and Table 5 showed that the proposed method marked the highest average SSIM. First, we discuss why the proposed method achieved the greatest evaluation. In particular, the score of the proposed method of the
soccer was much better than those of the conventional methods. In the soccer scene, large calibration errors cause several missing parts on a 3D model. Under such a situation, the proposed functions maintained the 3D model’s shape robustly to endure the camera calibration errors. In addition, in the soccer scene, it was difficult to extract a silhouette correctly because the color difference between an input image and a background Gaussian model is small due to shadow casting. In the proposed method, the error of silhouette extraction could be alleviated by the silhouette dilation. Although the C2F also partially improved the errors of silhouette extraction using an instance segmentation method, the errors could not be completely removed. Also, in the baseball scene, the evaluation values of all methods were approximately equal. The reason is that the camera calibration and the silhouette extraction were executed stably because the shooting environment was good, for example there were no shadow castings and few calibration errors. However, in the proposed method, the outlines of the reconstructed object were clearly reconstructed due to the effects of view-dependent shape refinement as shown in Fig. 11. In addition, the effects of the robustness on a calibration noise was observed in the reconstructed thin object such as a baseball bat as shown in Fig. 11(f).

In the SSIM evaluation, the condition #11 achieved better score than the condition #12 unlike the F-Measure evaluation. As the reason for this, we considered that missing parts on a 3D model were more noticeable than slight expanded parts. In addition, in the FVV system, a background 3D model such as a football pitch and an audience seat is prepared by a human modeler manually. Therefore, if an expanded part of a 3D model is correctly refined, the background 3D model becomes visible. At this time, the shape correctness causes bad SSIM scores because the color of the visible background 3D model does not match the color of the input image. Therefore, this mismatch caused the bad effect for the condition #12 because the mismatch of the condition #12 was bigger than that of the condition #11 due to higher threshold of the view-dependent refinement.

(2) Experiment 2-2 : Processing time comparison

Table 6 shows the detailed average processing time per frame of each method. In Table 6, silhouette extraction, 3D shape reconstruction, and rendering were implemented by GPU processing for all methods.

First, we describe the processing time of the silhouette extraction. As described in subsection 2.1, the C2F did not target the real-time processing of the silhouette extraction. In the method, an instance segmentation method called Mask R-CNN was introduced to improve the silhouette extraction quality. This instance segmentation method incurred much computation cost. In contrast, the FastP and the proposed method adopted the same silhouette extraction method based on a single Gaussian. The processing time of the proposed method was slightly longer than that of
the FastP because the proposed method adopted the adaptive dilation additionally. However, the increased calculation time was very small due to the simple processing.

Second, we explain the 3D shape reconstruction and rendering process. In the FastP, the 3D shape reconstruction and rendering processes are inseparable because the plane-based visual hull reconstruction is executed after a user selects a viewpoint for rendering. Also, the rendering process of the proposed method is the same as the C2F except for the function of view-dependent shape refinement. Although the rendering time of the proposed method was slightly slower than that of the C2F because of the refinement process, the increased processing time was very small.

As a result, the proposed method and the FastP ($P = 170$) achieved the real-time processing for both the soccer and the baseball. According to the results from Table 2 to Table 6, it was confirmed that the proposed method achieved both real-time processing and the highest accuracy. From these results, our proposed functions, the adaptive silhouette dilation and the view-dependent shape refinement, were effective to improve a real-time FVV quality in spite of the simplicity.

4.4 Discussion of the proposed method’s limitation

Next, we discuss the limitation of the proposed method.

(1) Effect of silhouette extraction errors

As shown in Fig. 11(c), the player’s legs in the light blue rectangle in the soccer scene are partially missing. In the region, the color difference between an input image and a background Gaussian model is small due to shadow casting. Therefore, the model quality degraded because of the errors of silhouette extraction using background subtraction.

To clarify the effect of these errors, we additionally experimented by using ground-truth silhouettes which were introduced in subsection 4.1. We used the ground-truth silhouettes for model creation based on the conditions #11 and #12 instead of the silhouettes generated by background subtraction. As a result, the average F-Measures of the conditions #11 and #12 using the ground-truth silhouettes were 0.895 (Soccer : 0.848, Baseball : 0.941) and 0.899 (Soccer : 0.862, Baseball : 0.937), respectively. Compared with the case using background subtraction in Table 2, the F-Measures of the conditions #11 and #12 using complete silhouettes were improved by 2.9% and 2.4%, respectively. Therefore, the additional investigation implies that the errors of silhouette extraction decreased the gain for improvement of the proposed method. In particular, the silhouette dilation scheme assumes that the complete silhouettes are input to the system. Therefore, further investigation is needed regarding the robustness of this issue. For example, the investigation of how silhouettes containing various errors affect the results of the proposed method is needed.

In addition, Fig. 11(f) indicates that the baseball glove was partially missing by view-dependent shape refinement because the color of it was similar to that of the baseball ground. It was found that the proposed method may be inferior when the background and the foreground colors are similar near 3D model outlines. Although the proposed method acquired high evaluation scores compared with the conventional methods, more accurate silhouette extraction and view-dependent refinement schemes are required for further improvement.

(2) Effect of errors caused by silhouette dilation

Also, the proposed method used landmarks set up
on the ground for calibrating a camera and for determining the silhouette dilation size. However, the target which is reconstructed as a 3D model rises up from the ground plane. Therefore, this discrepancy caused by the height difference may cause errors in order to refer to the calibration accuracy on the ground plane. In our experiments, we assume that the discrepancy does not become large because the distance between a target object and a real camera is large. To clarify this assumption, we additionally experimented using a goalpost in a soccer game. Using the camera parameter estimated by landmarks on the ground only, the error vector of the errors will be significant for specific objects such as a ball due to the height. In addition, this experiment was limited to the goalpost nearest region because there were no other landmarks whose height was known accurately. Therefore, the impact of the discrepancy needs to be analyzed using other shooting environments in the future.

5. Conclusion

In this paper, we proposed an accurate and real-time FVV synthesis algorithm. Our method uses adaptive silhouette dilation and view-dependent shape refinement to improve the quality of a reconstructed 3D model. First, the adaptive silhouette dilation focuses on reducing the missing part on a 3D model. Next, an expanded part caused by the silhouette dilation is scraped by the view-dependent shape refinement. The two functions are simple, but extreme quality improvement was observed in the experiments.

In the future, we will evaluate various sports scenes to confirm the robustness of the proposed method. In addition, in order to achieve further quality improvement, we will propose a more accurate silhouette extraction method which can process in real-time for many high-resolution images. Furthermore, we will improve the way of view-dependent refinement to reduce the missing part such as a baseball glove in the experiment. For example, we will devise how to automatically determine the optimal threshold $T_R$ and the filter size $a$ considering the amount of silhouette dilation.

References

1) A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand: ”3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards,” 2006 IEEE International Conference on Multimedia and Expo. 2161–2164, 2006.
2) C.-C. Lee, A. Tabatabai, and K. Tashiro: ”Free viewpoint video (FVV) survey and future research direction,” APSIPA Transactions on Signal and Information Processing, 4, E15, 2015.
3) A. Ishikawa, M. Teratani, S. Naito, S. Sakazawa, and A. Koike: ”Free viewpoint video generation for walk-through experience using image-based rendering,” ACM Multimedia 2008, 1007–1008, 2008.
4) N. Inamoto and H. Saito: ”Free Viewpoint Video Synthesis and Presentation from Multiple Sporting Videos,” 2005 IEEE International Conference on Multimedia and Expo, 2005.
5) O. Grau , G. A. Thomas, A. Hilton, J. Kilner, and J. Starck: ”A Robust Free-Viewpoint Video System for Sport Scenes,” 3DTV Conference 2007, 1–4, 2007.
6) T. Maeda, R. Suenaga, K. Suzuki, M.P. Tehrani, K. Takahashi, and T. Fujii ”Free Viewpoint Video for Sport Events Using Multi-resolution Visual Hull and Micro-facet Boarding,” International Workshop on Smart Info-Media Systems in Asia, SS 3-5, 191–196, 2016.
7) J. Chen, R. Watanabe, K. Nonaka, T. Konno, H. Sankoh, and S. Naito: ”Fast Free-viewpoint Video Synthesis Algorithm for Sports Scenes.” 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3209-3215, 2019.
8) T. Matsuyama, X. Wu, T. Takai, and T. Wada: ”Real-time dynamic 3-D object shape reconstruction and high-fidelity texture mapping for 3-D video,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 3, pp. 357–369, 2004.
9) K. Nonaka, R. Watanabe, J. Chen, H. Sabirin, and S. Naito: ”Fast plane-based free-viewpoint synthesis for real-time live streaming,” 2018 IEEE Visual Communications and Image Processing, 1–4, 2018.
10) K. Hayashi and H. Saito: ”Synthesizing Free-viewpoint Images from Multiple View Videos in Soccer Stadium,” International Conference on Computer Graphics, Imaging and Visualisation, 220-225, 2006.
11) H. Sankoh, S. Naito, K. Nonaka, H. Sabirin, and J. Chen: ”Robust Billboard-based, Free-viewpoint Video Synthesis Algorithm to Overcome Occlusions under Challenging Outdoor Sport Scenes,” ACM Multimedia 2018, 1724–1732, 2018.
12) T. Koyama, I. Kitahara and Y. Ohta, ”Live mixed-reality 3D video in soccer stadium,” The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality, 178–186, 2003.
13) Y. Ohta, I. Kitahara, Y. Kameda, H. Ishikawa, and T. Koyama, ”Live 3D Video in Soccer Stadium”, International Journal of Computer Vision, vol. 75, No. 1, 173–187, 2007.
14) K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz: ”Soccer on Your Tabletop,” Conference on Computer Vi-
Video 1 Quality comparison between the proposed method (#12) and the conventional methods, C2F ($V_F = 2cm$) and FastP ($P = 800$).

"A comparison between two 3D free-viewpoint generation methods: Player-billboard and 3D reconstruction," 2010 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, 1–4, 2010.

28) Q. Yao, H. Sankoh, K. Nonaka, and S. Naito: "Automated Camera Self-Calibration for Immersive Navigation of Free Viewpoint Sports Video," 2016 IEEE International Conference on Multimedia Signal Processing, 2016.

29) H. Tsurusaki, K. Nonaka, R. Watanabe, T. Konno, and S. Naito: "Sports Camera Calibration Using Flexible Intersection Selection and Refinement," ITE Transactions on Media Technology and Applications, vol. 9, No. 1, 95–104, 2021.

30) K. He, G. Gkioxari, P. Dollár, and R. Girshick: "Mask R-CNN," 2017 International Conference on Computer Vision, 2980–2988, 2017.

31) Z. Zhang: "A Flexible New Technique for Camera Calibration," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, No. 11, pp. 1330–1344, 2000.

32) L. Wang, F.C. Wu, and Z.Y. Hu: Multi-camera Calibration with One-dimensional Object under General Motions, IEEE International Conference on Computer Vision, 2007.

33) H. Shirahido and I. Kitahara, "Calibration of Multiple Sparsely Distributed Cameras Using a Mobile Camera", Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology, vol. 234, No. 1, 37–48, 2019.

34) Q. Yao, H. Sankoh, H. Sabirin, and S. Naito: "Accurate silhouette extraction of multiple moving objects for free viewpoint sports video synthesis," IEEE Workshop on Multimedia Signal Processing 2015, pp. 1–6, 2015.

35) W. Lorensen and H. Cline: "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," ACM SIGGRAPH Computer Graphics, vol. 21, 163–169, 1987.

36) "Nagoya University Multi-view Sequences Download List", http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/

37) R. Suenaga, K. Suzuki, T. Tsuchiya, M. P. Tehrani, K. Takahashi, and T. Fujii: "Practical Implementation of Free Viewpoint Sports Video Synthesis," ACM SIGGRAPH Computer Graphics, vol. 16, No. 2, 150–162, 1994.

38) K. He, G. Gkioxari, P. Dollár, and R. Girshick: "Mask R-CNN," 2017 International Conference on Computer Vision, 2980–2988, 2017.

39) Z. Zhang: "A Flexible New Technique for Camera Calibration," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, No. 11, pp. 1330–1344, 2000.

40) L. Wang, F.C. Wu, and Z.Y. Hu: "Multi-camera Calibration with One-dimensional Object under General Motions," IEEE International Conference on Computer Vision, 2007.

41) H. Shirahido and I. Kitahara, "Calibration of Multiple Sparsely Distributed Cameras Using a Mobile Camera", Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology, vol. 234, No. 1, 37–48, 2019.

42) "Nagoya University Multi-view Sequences Download List", http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/

43) Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, 600–612, 2004.
Ryosuke Watanabe received his B.E., M.E., and Ph.D degrees in Information Science from Hokkaido University in 2014, 2016, and 2020, respectively. He joined KDDI Corporation in 2016. He is currently a research engineer working in the Robotics Laboratory in KDDI Research, Inc. His research interests include free-viewpoint video, computer-generated holograms.

Tomoaki Konno received B.E. and M.E. degrees from Tohoku University in 2007 and 2009, respectively. He joined KDDI Corporation in 2009. He is currently a research engineer of the Robotics Laboratory in KDDI Research, Inc., Japan. His research interests include free viewpoint video and virtual reality. He is mainly engaged in system architecture design and development in free viewpoint video.

Keisuke Nonaka received his Ph.D degree in Information Processing from Tokyo Institute of Technology, Japan, in 2014. He joined KDDI Corporation in 2014 and is a research engineer working in image processing and computer vision. He was also a visiting scholar at the University of Southern California and was engaged in the research of graph signal processing in 2019. His research interests include free viewpoint video, virtual reality, and 3D model rendering.

Tatsuya Kobayashi received the B.E. and M.E. degrees in electronics engineering from Keio University in 2008 and 2010, respectively. In 2010, he joined KDDI Co. Ltd. His research interests include image processing and cross reality (XR). He is a R&D manager of Robotics Laboratory at KDDI Research, Inc.

Haruhisa Kato received B.E. and M.E. degrees in electronical and electronic engineering from Kobe University in 1997 and 1999, respectively. He received a Ph.D. in graduate school of engineering from Tokyo Institute of Technology in 2009. In 1999, he joined KDD Co. Ltd. His research interests include image processing and augmented reality. He is a Senior Expert of Robotics Laboratory at KDDI Research, Inc.

Masaru Sugano received the B.E. degree in electronics and communications engineering and the M.S. degrees in electronics, information and communications engineering from Waseda University in 1995 and 1997, respectively. Since 1997 he has been with Research and Development Laboratories of KDDI, and engaged in research on video coding and video/audio analysis for content-based search and retrieval. Currently he is a Senior Manager of Robotics Laboratory at KDDI Research, Inc.