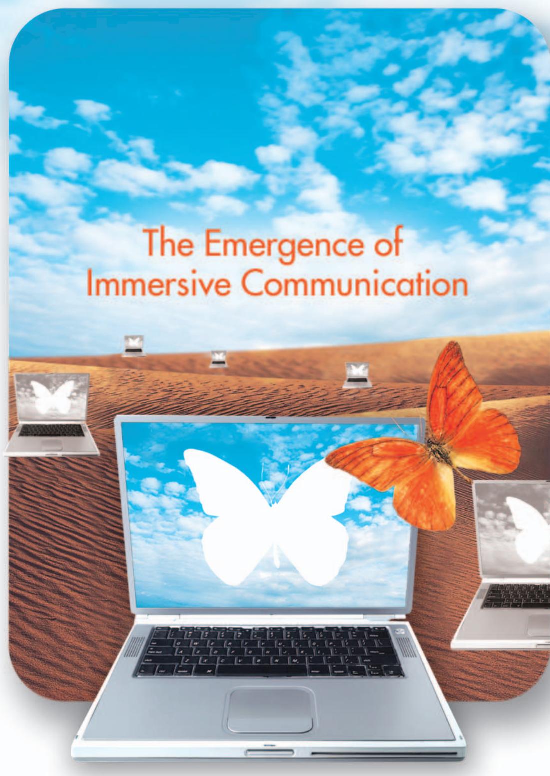
Masayuki Tanimoto, Mehrdad Panahpour Tehrani, Toshiaki Fujii, and Tomohiro Yendo

Free-Viewpoint TV

A review of the ultimate 3DTV and its related technologies



Free-viewpoint television (FTV) is an innovative visual media that enables us to view a three-dimensional (3-D) scene by freely changing our viewpoints. We proposed the concept of FTV and constructed the world's first real-time system including the complete chain of operation from image capture to display. We also carried out the FTV on a single personal computer (PC) and a mobile player. FTV is based on the ray-space method that represents one ray in real space with one point in the ray-space. We have developed several types of ray capture systems and interfaces such as a 360° capture/ray-reproducing display. FTV is regarded as the ultimate 3DTV, since it can generate infinite number of views. Thus, FTV is the key to immersive communication. Regarding FTV as the most challenging 3-D media, the Motion Picture Experts Group (MPEG) has been conducting its international standardization activities. This article reviews FTV and its related technologies.

INTRODUCTION

Television realized the human dream of seeing a distant world in real time. However, current TV provides us only a single view of a 3-D world and users cannot control their viewpoint. This is quite different from what we experience in the real world. 3DTV has more views. It gives us depth feeling. Although stereoscopic 3DTV has only two views, autostereoscopic 3DTV has three or more views and realizes motion parallax. However, the viewing zone of autostereoscopic 3DTV is very limited. A multiview system sets many cameras in a large 3-D space and displays captured views by switching them. It realizes motion parallax with a large viewing zone. However, the number of viewpoints is fixed in the multiview system. In all visual systems explained above, users cannot set their viewpoints freely. If we can move our viewpoints freely in the 3-D space, it would greatly contribute to immersive communication. In the field of computer graphics, we have the model of 3-D scene and can generate free-viewpoint images easily in a virtual world. However, free-viewpoint image generation in a real world is very difficult.

We challenged it and succeeded to realize FTV [1]–[3]. FTV enables us to view a 3-D scene by freely changing the viewpoint. We have proposed the concept of FTV and constructed the world's first real-time system [3] including the complete chain of operation from image capture to display. FTV is the ultimate 3DTV since it can generate infinite number of views, thus, FTV is the key to immersive communication.

To realize FTV, the following problems should be solved:

- *Representation*: We have to describe all views at any viewpoint in 3-D space efficiently. Suitable data representation differs in various multiview/free-viewpoint systems.
- *Capturing*: Many cameras are used to capture 3-D scene. We need to treat cameras with different characteristics as if they were a single camera.

Digital Object Identifier 10.1109/MSP.2010.939077
Date of publication: 17 December 2010

- *Rendering*: We can capture only finite number of views. Then, we have to generate free viewpoint image using the limited number of views.
- *User Interface*: FTV can generate infinite number of views; however human eyes can only see one to two views. Thus, user interface is needed to adapt FTV output to human eyes.
- *Coding*: The captured data of FTV is huge. Therefore, an efficient compression scheme should be developed.

DATA REPRESENTATION OF A 3-D SCENE

There are various ways of data representations of 3-D scene. They are discussed in the view-geometry domain as shown in Figure 1. In the figure, the horizontal axis denotes the number of views "N," where views are images captured by a multicamera system from a 3-D scene. The vertical axis denotes the number of depths "M," where depth means depth map. A depth map is an image in which the intensity of each pixel represents distance between the optical center of a camera and the surface of an object in 3-D scene. Depth corresponds to the geometrical information. Many depths can be represented in different form, e.g., a 3-D model. A camera position has view and/or depth. It means number of views and depths can be different in a multicamera setting. When M = N, all camera positions have a view and a depth. However, if N > M, there are some camera positions that have view, but no depth. On the other hand, for M > N, there are some camera positions that have depth, but no view. A general form of view-geometry representation is multiview plus multidepth (N View + M Depth) that covers all other presentations. Details about different representations will be given as follows.

FREE-VIEWPOINT TELEVISION IS AN INNOVATIVE VISUAL MEDIA THAT ENABLES US TO VIEW A THREE-DIMENSIONAL SCENE BY FREELY CHANGING OUR VIEWPOINTS.

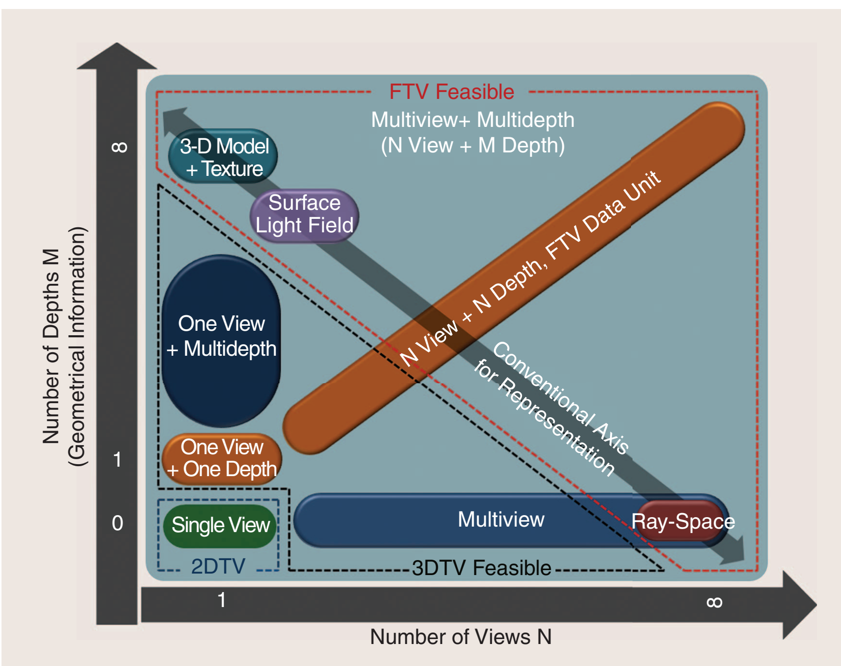
Representations with no geometrical information are as follows. The conventional two-dimensional (2-D) TV (2DTV) system is achieved by a single view image. Representation with more view than one view is multiview representation. If the number of views is very large, multiview representation is equivalent to ray-space [4]–[6], which will be explained in the next section.

Representations in the following are for the case of having only one view with variable geometrical information. Using one view plus one depth representation we can achieve a free-viewpoint image generation in a narrow range. By adding more geometrical information or having multidepth, we can represent the free-viewpoint system as one view plus multidepth, such as layered-depth images (LDI) [7]. LDI contains more geometrical information, so that it can provide wider range of free-viewpoint. With all geometrical information or a 3-D model of the scene, we have a 3-D model plus texture representation that can provide a wide range of free-viewpoint.

Remaining representations are as follows. Surface light field [8] contains less geometrical information in comparison with the 3-D model, while several views are required for this representation. This representation can provide wide range of free-viewpoint images. Another representation is (N View + N Depth) that is considered by MPEG. It covers a large area of the view-geometry representation. Based on this representation, we also proposed the FTV Data Unit (FDU) [9], which has a compact structure for efficient representation of (N View + N Depth). FDU consists of view, depth, additional information for right view synthesis, and additional information for left view synthesis.

In [10], various 3-D scene representations are surveyed in one dimension, in which the more geometrical information corresponds to the less number of view and vice versa. As shown in Figure 1, it is along the axis connecting the upper-left point, i.e., 3-D model/texture, to the far right-bottom point, i.e., ray-space.

Different representations in Figure 1 can be grouped into feasible applications. The region bounded with a red dashed line is feasible for FTV implementation. A dashed black line is the area where the 3DTV is feasible. We realized FTV using



[FIG1] **View-geometry representation.**

multiview representation when the views are sufficiently dense. Since the multiview representation is located in the dashed black line, we obtained depth information from multiview images to realize FTV.

FTV IS THE ULTIMATE 3DTV SINCE IT CAN GENERATE INFINITE NUMBER OF VIEWS, THUS, FTV IS THE KEY TO IMMERSIVE COMMUNICATION.

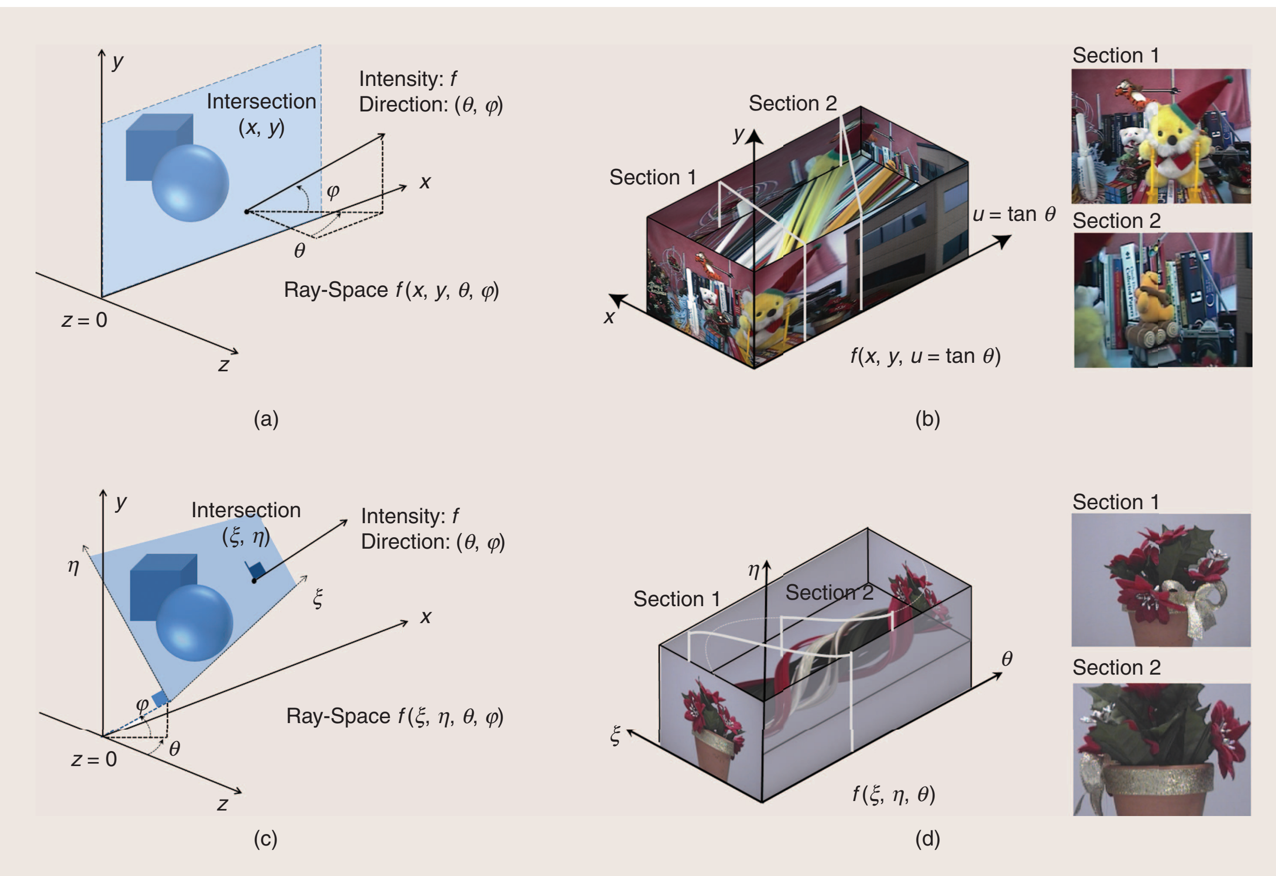
RAY-SPACE REPRESENTATION

Ray-space [4]–[6] was proposed to describe rays in a 3-D space. It can be a common data format for 3-D visual communication. We developed FTV based on the ray-space representation. In ray-space representation, one ray in the 3-D real space is represented by one point in the ray-space. The ray-space is a virtual space. However, it is directly connected to the real space. The ray-space is easily generated by collecting multiview images while giving consideration to the camera parameters.

Let (x, y, z) be three space coordinates, and (θ, φ), where $-\pi/2 \le \theta \le \pi/2$, $-\pi/2 \le \varphi \le \pi/2$, be the parameters of direction as shown in Figure 2(a) for the orthogonal ray-space. These ray parameters construct a five-dimensional (5-D) ray-space $(x, y, z, \theta, \varphi)$ of multiview images. We define a function $f(x, y, z, \theta, \varphi)$ whose value corresponds to a intensity of a ray. Although the 5-D ray-space mentioned above includes all information viewed from any viewpoint, it is highly redundant due to the straight traveling paths of the rays. Thus, when we treat rays that arrive at a reference plane, we can reduce the dimension of the parameter space to four-dimensions. We call this ray-parameter space the "ray-space."

Although the parameterization is simple in orthogonal ray-space, this parameterization cannot represent all rays; for example, rays parallel to plane $z = 0$ $(\varphi = \pi/2)$. Thus, spherical ray-space shall be defined as shown in Figure 2(c). Given a plane with the normal vector in the direction of (θ, φ) where $-\pi \le \theta \le \pi$, $-\pi/2 \le \varphi \le \pi/2$, and coordinate of (ξ,η), the ray emitting from the surface of an object perpendicular to the plane at intersection of (ξ,η) is defined by $f(\xi,\eta,\theta,\varphi)$.

The following explains how ray-space is captured, constructed, and how free-viewpoint images are generated. For $\varphi = 0$, the orthogonal and the spherical ray-spaces can be captured by using linear and circular camera arrangements, respectively.



[FIG2] Ray-space representation and free-viewpoint image generation.(a) Definition of orthogonal ray-space. (b) Example of orthogonal ray-space and rendering for linear camera arrangement ($\phi = 0$). (c) Definition of spherical ray-space. (d) Example of spherical ray-space and rendering for circular camera arrangement ($\phi = 0$).

Ray-space is constructed by transforming captured images by cameras in the real world into ray-space domain, in which transformed images are aligned in parallel. The image planes in the orthogonal ray-space are flat planes whereas they are a sine-shaped plane in the spherical ray-space. Figure 2(b) depicts examples of the constructed ray-space for the orthogonal ray-space and spherical ray-space when $\varphi = 0$. Constructed ray-space can be used for free-viewpoint image generation. Given the location of virtual viewpoint, a free-viewpoint image can be generated by cutting the ray-space vertically. As illustrated in Figure 2(d), in the orthogonal ray-space, the vertical cross-section is a flat plane whereas in spherical ray-space, it is a sine-shaped plane. If the captured images are not dense enough for free-viewpoint image generation, ray-space interpolation should be performed to obtain a large number of views. In this process, virtual views are placed between real camera views as if the camera arrangement is very dense.

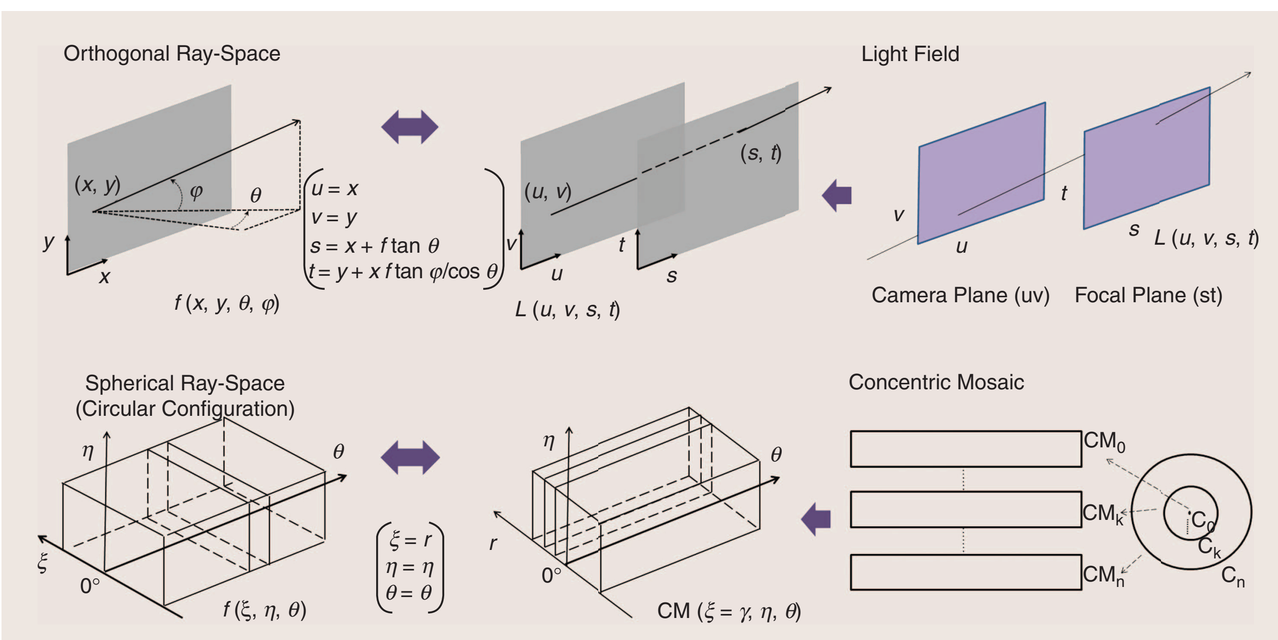
THE RAY-SPACE IS EASILY GENERATED BY COLLECTING MULTIVIEW IMAGES WHILE GIVING CONSIDERATION TO THE CAMERA PARAMETERS.

After ray-space, light field rendering [11] and concentric mosaic [12] were proposed. Light field and concentric mosaic represent rays captured by using linear and circular camera arrangements, respectively. Light field is the same as orthogonal ray-space, and concentric mosaic is the same as spherical ray-space. The set of rays in a light field can be represented using two-plane parameterization (u, v) and (s, t). Similarly, orthogonal ray-space is also represented with four parameters (x, y) and (θ, φ) in which $u = x, v = y$, $s = x + f\tan\theta$, and $t = y + f\tan\varphi/\cos\theta$. Concentric mosaics index all input image rays with three parameters: radius r, rotation angle θ, and vertical elevation η. If unfolded concentric mosaic (CM_i) images are aligned in parallel, they are equivalent to spherical ray-space for circular configuration ($\varphi = 0$), in which $\xi = r, \eta = \eta, \theta = \theta$. Figure 3 illustrates how the ray-space method includes both the light field and concentric mosaic.

THE FTV SYSTEM

The FTV system contains capturing, correction, rendering, user interface, and coding components. The ray-space interpolation described in the section "Ray-Space Representation" consists of depth estimation and interpolation units. Depth estimation is used to find lines in ray-space, while interpolation is used to generate the intensity values along the line. Figure 4 shows three FTV system architectures, based on the location of depth estimation and interpolation.

- *Case A*: Both depth estimation and interpolation units are at the receiver. Multiview images are sent to the receiver. Rendering that includes depth estimation and interpolation is performed for the desired virtual viewpoint. Due to high computational cost of depth estimation process, this case requires high processing power at the receiver. Based on this architecture, we developed a real-time FTV system using a PC cluster [3], [13], [14], since Case A is the straightforward architecture for rendering free-viewpoint images. Currently, we can generate free-viewpoint images in real time on a single PC or a mobile player. Coding in this architecture is Multiview Video Coding (MVC). MVC was standardized by MPEG and will be explained briefly in the section "International Standardization of FTV."



[FIG3] Ray-space includes light field and concentric mosaic.

- *Case B*: Depth estimation is performed at the sender whereas interpolation is performed at the receiver. Corresponding depth is estimated for each view images at the sender, independent of the desired viewpoint. Multiview + depth are sent to the receiver. In this case, rendering at the receiver is only view interpolation. However, the interpolation of Case B is different from that of Case A. Details will be explained in the section "Rendering." This case has a light processing task at the receiver since the depth estimation that requires high processing power is performed at the sender side.
- *Case C*: Both depth estimation and interpolation are performed at the sender. Ray-space is reconstructed at sender, and sent to the receiver. The rendering in this case is only to select pixels from the ray-space by the user interface. Since both depth estimation and interpolation are performed at the sender, the processing task is very high at the sender while it is very light at the receiver. Coding in this case is a challenging research issue since ray-space has a huge amount of data.

THE FTV SYSTEM CONTAINS CAPTURING, CORRECTION, RENDERING, USER INTERFACE, AND CODING COMPONENTS.

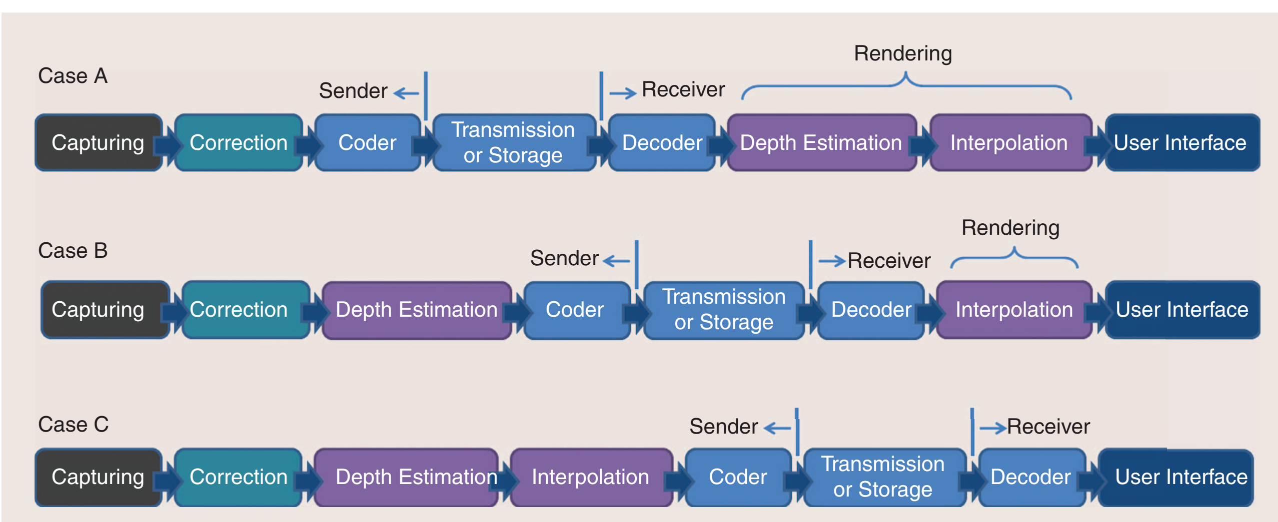
CAPTURE
The following discusses our developed capturing systems, such as a multicamera system with different configurations, and a mirror-scan ray capturing system.

We constructed a one-dimensional-arc capturing system for a real-time FTV system [13], [14]. It consists of 16 cameras, 16 clients, and one server. Each client has one camera and all clients are connected to the server with gigabit Ethernet. We further improved the system to capture a 3-D scene by 100 cameras. Figure 5(a) shows the "100-camera system" [15]. This system consists of one host-server PC and 100 client PCs (called nodes) connected to each camera. The host PC generates a synchronization signal and distributes it to all of the nodes. This system is capable of capturing not only high-resolution video with 30 frames/s, but also analog signals of up to 96 kHz. The camera setting is flexible as shown in Figure 6(a). MPEG test sequences "Rena," "Pantomime," "Champagne_tower," "Dog," "Kendo," and "Balloons" were taken using the 100-camera system. Due to different characteristic of cameras, geometrical and color correction should be performed after capturing.

We have also developed a 360° mirror-scan ray capturing system [16] as shown in Figure 5(b). This system uses two parabolic mirrors. Incident rays that are parallel to the axis of a parabolic mirror gather at the focus of the parabolic mirror. Hence, rays that come out of an object placed at the focus of the lower parabolic mirror will gather at the focus of the upper parabolic mirror. Then, the real image of the object is generated at the focus of the upper parabolic mirror. A rotating aslope mirror scans rays at the focus of the upper parabolic mirror. Finally, the image from the aslope mirror is captured by a high-speed camera. By using this system, we can capture all-around convergent views of an object.

CORRECTION
In a multicamera system, the captured images contain the misalignment and luminance differences of the cameras. In this process, camera calibration can be performed offline whereas the corrections can be performed in real time.

For geometry correction, we have developed the camera-calibration-based rectification method. Many geometry correction approaches have the limitation of number of views or cameras. Our geometry correction method is designed for a multicamera setup, where any number of cameras can be corrected together. The outline of our correction method is as follows. First, a chessboard pattern is used as a



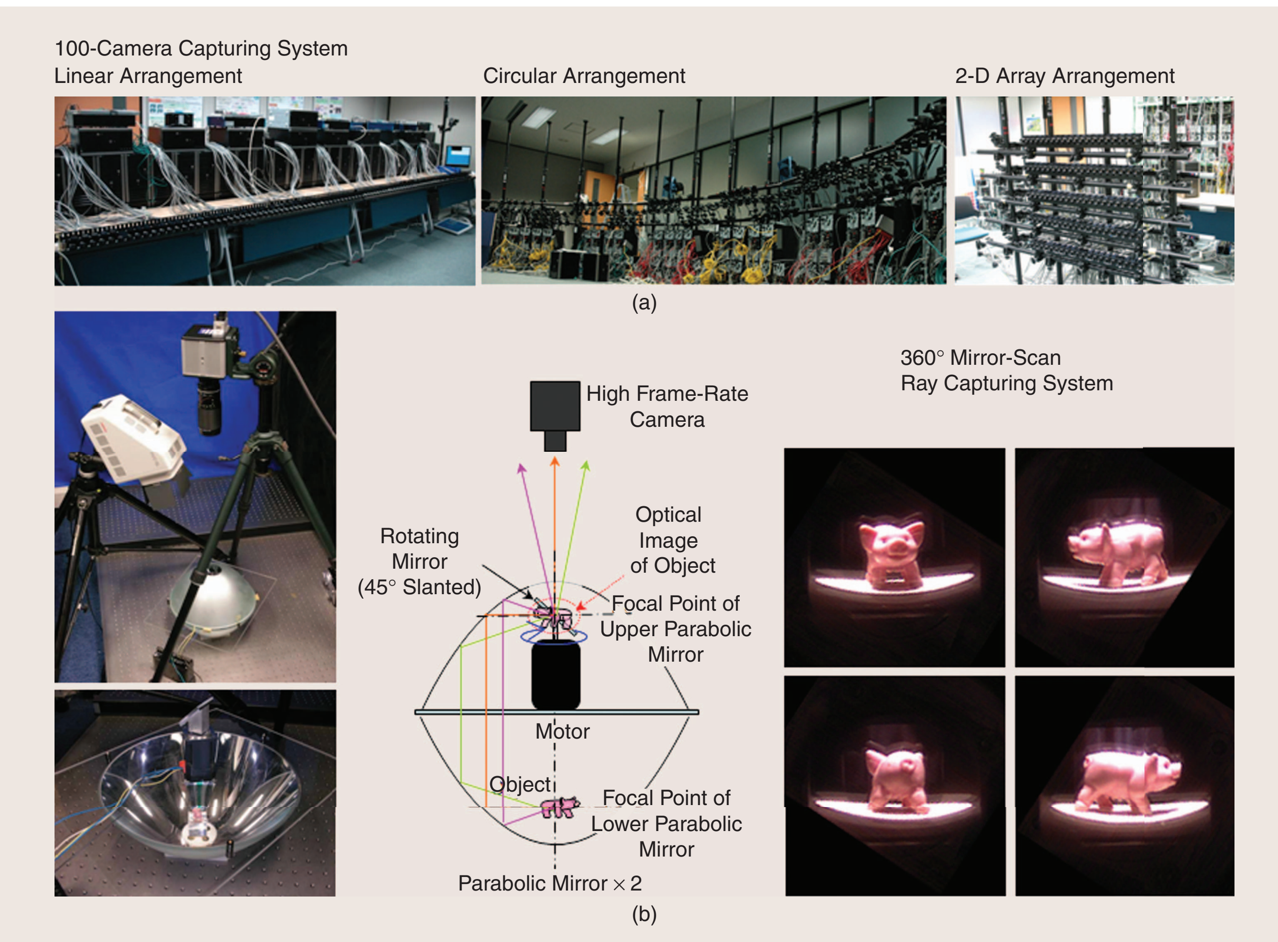
[FIG4] **FTV architectures.**

calibration rig, where corner points are detected and used for estimation of intrinsic, extrinsic, and lens distortion parameters. In this step, all cameras are calibrated [17] and lens distortions are removed. In Figure 6(a), detected corner points of the chessboard are shown with green dots. Second, using the initial position/orientation of actual cameras (i.e., shown in magenta) and position of corner dots, an ideal camera array (i.e., shown in magenta) is fitted by principal component analysis (PCA) as shown in Figure 6(a). Finally, we correct homographies by computing the relationship between ideal projection and a real captured image. Note that our geometry correction does not need camera intrinsic and extrinsic parameters. However, we can provide the camera parameters in complete form; unlike the non-metric rectification methods that perform correction after capturing without camera calibration.

CODING OF FTV DATA IS A CHALLENGING AND IMPORTANT ISSUE THAT HAS BEEN FOCUSED ON BY MPEG.

For color correction, our method minimizes the total error from target color space. First, we obtained a color correspondence map by using a "Macbeth" color chart. To obtain a matched color vector, we have to capture the color pattern at different exposure times. Then, we can obtain a color correction matrix that minimizes the error using iteration-based optimization. The iteration is stopped when the amount of error becomes small. Our approach keeps the consistency of color in all cameras. It is different from correction methods that mostly focus on the relationship between the nearest two cameras. In these methods, the color distance among neighborhood cameras stays sufficiently small. However, the distance between cameras at end-to-end might be large because errors at each camera are accumulating. Figure 6(b) shows the result of geometry and color correction on a test sequence.

In the 360° mirror-scan ray capturing system, the captured images have distortion caused by the parabolic mirror optics and, rotation-distortion caused by the rotation of the aslope mirror. To compensate for these, simple 2-D image transformation should be done for each frame of the high-speed camera.



[FIG5] **Parts (a) and (b) show 100-camera and 360° mirror-scan ray capturing systems.**

RENDERING

We developed high-quality rendering using global optimization methods for depth estimation, such as loopy belief propagation [18] and multipass dynamic programming [19]. Within rendering, original images, which generally come from the left and right sides, are defined as the references to generate a virtual view. Rendering in Case A [18]–[20], Case B [21]–[25], and Case C of FTV architecture in Figure 4 are different.

FTV IS THE MOST ADVANCED VISUAL SYSTEM, AND IT WILL BRING AN EPOCHAL CHANGE IN THE HISTORY OF TELEVISION.

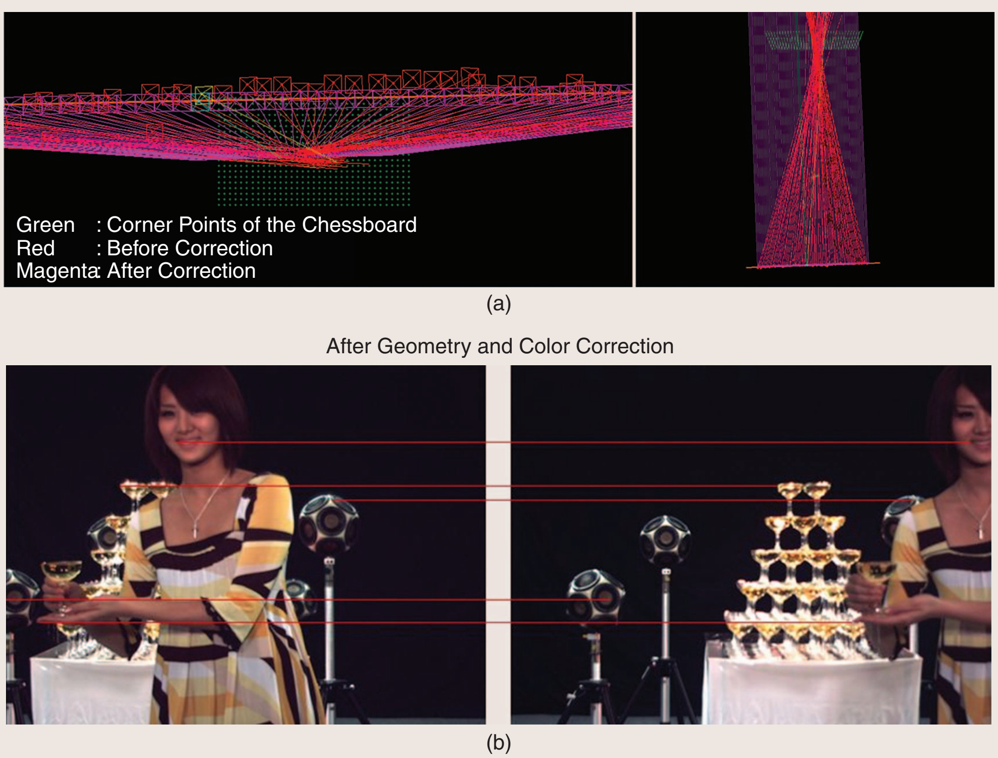
In Case A, we have mutliview images as reference views at the receiver. Given a desired location of virtual viewpoint, two nearest reference views are determined. Then, stereo matching is performed to find the geometrically corresponding locations in the reference views. It is the task of the depth estimation unit. Finally, found corresponding locations are interpolated to generate the image at the location of the virtual viewpoint.

In Case B, the synthesis framework uses multiview plus depth. The corresponding depth for each view is estimated at the sender. Given the desired virtual viewpoint, and the two nearest reference views plus depths, the virtual view is rendered by interpolating the warped reference views to the location of the virtual view using their depth maps. Occluded regions in one reference view are filled by the other reference view if they are visible there. Cracks, holes and remaining occluded regions that cannot be seen in both references are filled using the image processing method, such as filtering and inpainting.

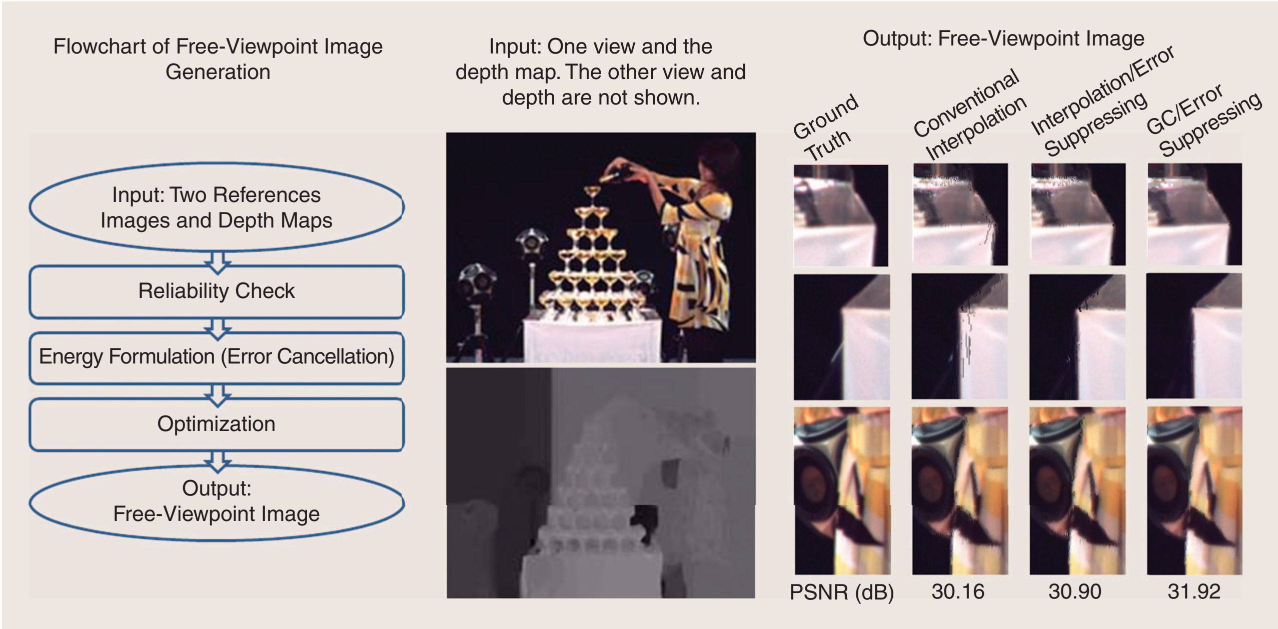
In Case C, rendering is extremely easy, and it is equivalent to pixel selection of the ray-space, as shown in "Ray-Space Representation."

In all cases, the synthesized view contains artifacts if there is error in the depth estimation. Considering the fact that synthesis artifacts can be measured at the location of the reference views, we can eliminate similar errors in the synthesized location. Based on this idea, we have developed two methods that can be applied during free-viewpoint rendering to suppress errors in occluded and nonoccluded regions.

The first method [23] estimates the residual errors at the virtual viewpoint and subtracts them from the synthesized view. The virtual image at one camera position is generated in the other camera position. The residual error between the original and generated view is used to estimate the residual errors in the position of the synthesized view. This method is effective for nonoccluded regions and improves peak signal-to-noise ratio (PSNR). In the second method [24], camera views are projected to each other and the differences between original and generated views are obtained. They are used for a reliability check. Unreliable pixels from one camera view are replaced by the reliable pixels from the



[FIG6] **Parts (a) and (b) show the correction of geometry and color in FTV.**



[FIG7] **Free-viewpoint image generation using two reference views and depth maps.**

other camera view. This method is effective for occluded regions and eliminates artifacts. The combination of the two methods can provide improvement in comparison with the conventional method as shown in Figure 7 at both, eliminating artifacts and improving PSNR, respectively. The combined method approaches the view synthesis as a reliable interpolation or energy optimization problem using graph cuts (GC) [25].

USER INTERFACE

To adapt FTV output to human vision, we have developed two types of user interface for FTV. Figure 8 shows the FTV on different platforms and user interfaces.

The first type of user interface shows views only at the viewpoints given by the user. User's viewpoint can be given using an eye/head-tracking system, a remote controller, or a touch panel. We realized this type of user interface on conventional 2-D, stereoscopic, and autostereoscopic 3-D displays. For autostereoscopic 3-D displays, binocular parallax for the eyes are provided by the display while motion parallax is provided by head tracking and changing the image adaptively. Many head-tracking systems have been proposed using magnetic sensors, various optical markers, infrared cameras, and retroreflective light from retinas. Our head-tracking system uses only a conventional 2-D camera and detects the position of a user's head by image processing. The user does not need to attach any markers or sensors. In the user interface using a 2-D display, the location of the user's head is detected with the head-tracking system and the corresponding view image is generated.

The second type of user interface provides all views so that users can see any views by changing their locations in front of the interface. Seelinder [26] belongs to the second type. It is a 360°, ray-producing display that allows multiple viewers to see 3-D FTV images.

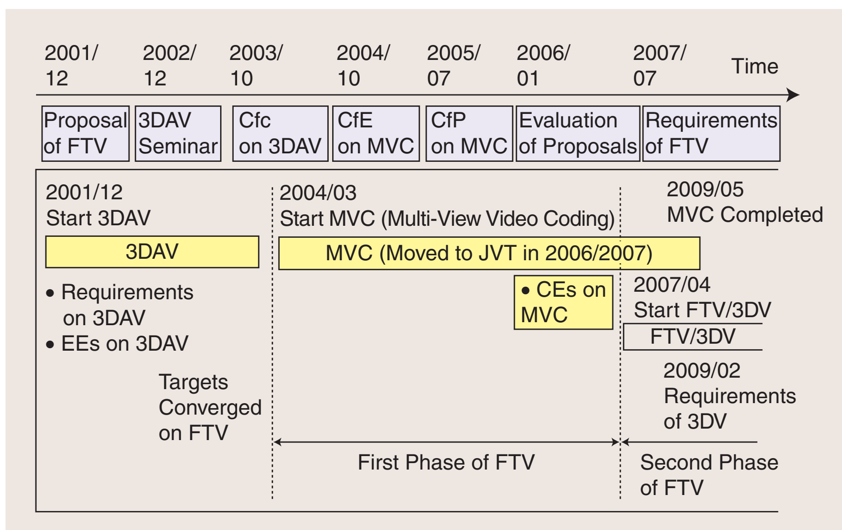
INTERNATIONAL STANDARDIZATION OF FTV

Coding of FTV data is a challenging and important issue that has been focused on by MPEG. To achieve an efficient and standard method for coding of multiview images, MPEG started with the standardization of FTV. The history of FTV standardization in MPEG is shown in Figure 9. We proposed FTV to MPEG in December 2001. At the first stage, many 3-D topics were discussed. The discussion was converged on FTV in January 2004. MPEG regarded FTV as the most challenging 3-D system and started the standardization of the coding part of FTV as MVC (Multiview Video Coding) [27]. In multiview video data, there is correlation among views. This redundancy can be removed by interview predication. It can also be done by using motion compensation that is used in conventional video coding. MVC applies the motion compensation method in H.264/MPEG-AVC to not only time but also view direction. MVC outperforms simulcast coding using H.264/MPEG-AVC by an average bitrate savings of 20–30%.

To discuss other issues on FTV, such as data format and view generation, MPEG-FTV was established in April 2007. MPEG-FTV defined the FTV reference model, where multiview plus the depth format is adopted. In January 2008, MPEG-FTV targeted the standardization of 3-D Video (3DV). The view generation function of FTV is used to decouple capture and display in 3DTV. 3DV is the second phase of FTV. This activity of MPEG-FTV is the standardization of a coding method for the Case B architecture of FTV in Figure 4. It enables 3-D display adaptation with different types, sizes, and viewing preferences, and supports high-quality

[FIG8] FTV on different platforms and user interfaces.

autostereoscopic displays by generation of many high-quality views from a limited amount of input data.



[FIG9] History of FTV Standardization in MPEG.

CONCLUSIONS

FTV is a ray-based system and contains advanced technologies of ray capture, processing, and display. Ray is the most essential element of visual systems. Thus, FTV is an innovative media that is adapted to the human visual system with natural interface between humans and the environment. FTV allows one to view a 3-D world by freely changing the viewpoint. It is the most advanced visual system and will bring an epochal change in the history of television.

FTV will find many applications in the fields of broadcast, communication, amusement, entertainment, advertising, exhibition, education, medicine, art, archives, security, and surveillance. It has already been realized in real time, and now we can enjoy FTV on a PC or a portable player. The international standardization of FTV was conducted at MPEG and will be completed in the near future. Therefore, the introduction of FTV is not far if the standards of FTV are available and its contents are delivered over the Internet or with packaged media.

ACKNOWLEDGEMENT

This work is partially supported by Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications, and National Institute of Information and Communications Technology, Japan.

AUTHORS

Masayuki Tanimoto (tanimoto@nuee.nagoya-u.ac.jp) received the B.E., M.E., and Dr.E. degrees in electronic engineering

from the University of Tokyo in 1970, 1972, and 1976, respectively. He joined Nagoya University in 1976. Since 1991, he has been a professor at the Graduate School of Engineering, Nagoya University. His research interests include image coding, image processing, 3-D imaging, FTV, and ITS. He is the former president of the Institute of Image Information and Television Engineers (ITE), and a fellow of the Institute of Electronics, Information, and Communication Engineers (IEICE) and ITE. He received the IEICE Achievement Award and the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology. He is a Senior Member of the IEEE.

Mehrdad Panahpour Tehrani (panahpour@nuee.nagoya-u.ac.jp) received the Dr.Eng. degree in information electronics from Nagoya University, Japan, in 2004. From 2004 to 2007, he was with the Information Technology Center, Nagoya University as a postdoctoral researcher. He was an associate research engineer in KDDI R&D Laboratories Inc. from 2007 to 2009. Currently, he is an associate professor at the Graduate School of Engineering, Nagoya University. His research interests are 3-D image processing and communications, multiview coding, distributed source coding in camera sensor networks, and 3-D media integration and communication. He is a Member of the IEEE, ITE, and IEICE.

Toshiaki Fujii (fujii@it.ss.titech.ac.jp) received the Dr.E. degree in electrical engineering from the University of Tokyo in 1995. From 1995 to 2007, he was with the Graduate School of Engineering, Nagoya University. He is currently an associate professor in the Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include multidimensional signal processing, large-scale multicamera systems, multiview video coding and transmission, FTV, and their applications for intelligent transport systems. He is a Member of the IEEE, IEICE, and the Institute of Image Information and Television Engineers of Japan.

Tomohiro Yendo (yendo@nuee.nagoya-u.ac.jp) received the B.E., M.E., and Ph.D. degrees from Tokyo Institute of Technology, Japan, in 1996, 1998, and 2001, respectively. From 1998 to 2002, he was a researcher for the Advanced 3-D Television Project at the Telecommunications Advancement Organization of Japan. From 2002 to 2004, he was a research fellow at Japan Science and Technology Agency. He is now an assistant professor at at the Graduate School of Engineering, Nagoya University. His current research interests include 3-D image display, capturing, and processing.

REFERENCES

[1] M. Tanimoto, "Free viewpoint television," (in Japanese), *J. Three Dimensional Images*, vol. 15, no. 3, pp. 17–22, Sept. 2001.

[2] M. Sekitoh, T. Fujii, T. Kimoto, and M. Tanimoto, "Bird's eye view system for ITS," in *Proc. IEEE IV Symp.*, 2001, pp. 119–123.

[3] M. Tanimoto, "Overview of free viewpoint television," *Signal Process. Image Commun.*, vol. 21, no. 6, pp. 454–461, July 2006.

[4] T. Fujii, "A basic study on integrated 3-D visual communication," (in Japanese), Ph.D. dissertation in engineering, Dept. Electrical Eng., Faculty of Eng., Univ. of Tokyo, Japan, 1994.

[5] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3D visual communication," in *Proc. Picture Coding Symp. 1996*, Mar. 1996, pp. 447–451.

[6] T. Fujii and M. Tanimoto, "Free-viewpoint television based on the ray-space representation," in *Proc. SPIE ITCom 2002*, Aug. 2002, pp. 175–189.

[7] J. Shade, S. J. Gortler, L. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH 98*, July 1998, pp. 231–242.

[8] W. C. Chen, J. Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 447–456, 2002.

[9] M. Tanimoto and M. Wildeboer, "Framework for FTV coding," in *Proc. Picture Coding Symp. (PCS2009)*, May 2009, vol. SS-2, 128.

[10] S. C. Chan, H.-Y. Shum, and K.-T. Ng, "Image-based rendering and synthesis," *IEEE Signal Processing Mag.*, vol. 24, pp. 22–33, Nov. 2007.

[11] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH (ACM Trans. Graphics)*, Aug. 1996, pp. 31–42.

[12] H. Y. Shum and L. W. He, "Rendering with concentric mosaics," in *Proc. SIGGRAPH ACM Trans. Graphics*, Aug. 1999, pp. 299–306.

[13] P. Na Bangchang, T. Fujii, and M. Tanimoto, "Experimental system of free viewpoint television," in *Proc. IST/SPIE Symp. Electronic Imaging*, vol. 5006–5066, pp. 554–563, Jan. 2003.

[14] T. Fujii and M. Tanimoto, "Real-time ray-space acquisition system," *SPIE Electron. Imaging*, vol. 5291, pp. 179–187, Jan. 2004.

[15] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound: 100-camera and microphone system," in *Proc. IEEE 2006 Int. Conf. Multimedia & Expo (ICME)*, July 2006, pp. 437–440.

[16] K. Manoh, T. Yendo, T. Fujii, and M. Tanimoto, "Ray-space acquisition system of all-around convergent views using a rotation mirror," in *Proc. SPIE Three-Dimensional TV, Video, and Display VI*, Sept. 2007, vol. 6778, pp. 67780C-1–67780C-8.

[17] Z. Zhang, "A flexible new technique for camera calibration," Microsoft Research, Microsoft Corporation, Tech. Rep. MSR-TR-98-71, Dec. 2, 1998.

[18] M. Droese, T. Fujii, and M. Tanimoto, "Ray-space interpolation constraining smooth disparities based on loopy belief propagation," in *Proc. IWSSIP 2004*, Poznan, Poland, Sept. 2004, pp. 247–250.

[19] N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "Real-time arbitrary view interpolation and rendering system using ray-space," in *Proc. SPIE Three-Dimensional TV, Video, and Display IV*, Nov. 2005, vol. 6016, pp. 250–261.

[20] N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "Free viewpoint image generation using multi-pass dynamic programming," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XIV*, Feb. 2007, vol. 6490, pp. 460–470.

[21] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI*, San Jose, CA, 2004, pp. 93–104.

[22] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Process. Image Commun.*, vol. 24, no. 1–2, pp. 65–72, Jan. 2009.

[23] H. Furihata, M. P. Tehrani, T. Yendo, T. Fujii, and M. Tanimoto, "Novel view synthesis with residual error feedback for FTV," in *Proc. IS&T/SPIE Electronic Imaging, Session on DIBR and FTV*, San Jose, CA, Jan. 2010, vol. 7524, pp. 75240K-1–75240K-12.

[24] L. Yang, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Artifact reduction using reliability reasoning for image generation of FTV," *J. Vis. Commun. Image Represent.*, vol. 21, no. 5–6, pp. 542–560, July/Aug. 2010.

[25] L. Yang, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Probabilistic reliability based view synthesis for FTV," in *Proc. IEEE Int. Conf. Image Processing (ICIP2010)*, Sept. 2010, pp. pp. 1785–1788.

[26] T. Yendo, T. Fujii, M. Tanimoto, and M. P. Tehrani, "The seelinder: Cylindrical 3D display viewable from 360 degrees," *J. Vis. Commun. Image Represent.*, vol. 21, no. 5–6, pp. 586–594, July/Aug. 2010.

[27] (2005, July). *Introduction to Multi-View Video Coding*. ISO/IEC JTC 1/SC 29/WG11, N7328

[SP]