Akira Kubota, Aljoscha Smolic, Marcus Magnor,
Masayuki Tanimoto, Tsuhan Chen,
and Cha Zhang

© BRAND X PICTURES

Multiview Imaging and 3DTV

Special issue overview and introduction

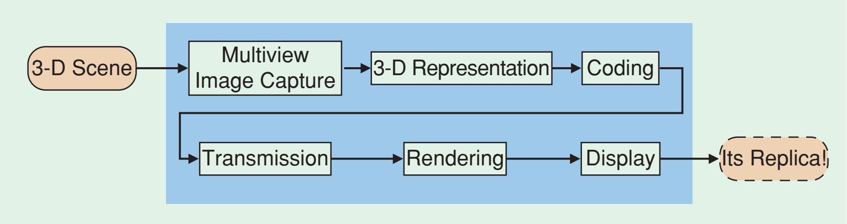
Multiview imaging (MVI) has attracted increasing attention, thanks to the rapidly dropping cost of digital cameras. This opens a wide variety of interesting new research topics and applications, such as virtual view synthesis, high-performance imaging, image/video segmentation, object tracking/recognition, environmental surveillance, remote education, industrial inspection, 3DTV, and free viewpoint TV (FTV) [9], [10]. While some of these tasks can be handled with conventional single view images/video, the availability of multiple views of the scene significantly broadens the field of applications, while enhancing performance and user experience.

3DTV and FTV are some of the most important applications of MVI and are new types of media that expand the user experience beyond what is offered by traditional media. They have been developed by the convergence of new technologies from computer graphics, computer vision, multimedia, and related fields. 3DTV, also referred to as stereo TV, offers a three-dimesional (3-D) depth impression of the observed scene, while FTV allows for an interactive

Digital Object Identifier 10.1109/MSP.2007.905873

1053-5888/07/$25.00©2007IEEE

selection of viewpoint and direction within a certain operating range. 3DTV and FTV are not mutually exclusive. On the contrary, they can be very well combined within a single system as they are both based on a suitable 3-D scene representation. In other words, given a 3-D representation of a scene, if a stereo pair of images corresponding to the human eyes can be rendered, the functionality of 3DTV is provided. If a virtual view (i.e., not an actual camera view) corresponding to an arbitrary viewpoint and viewing direction can be rendered, the functionality of FTV is provided.



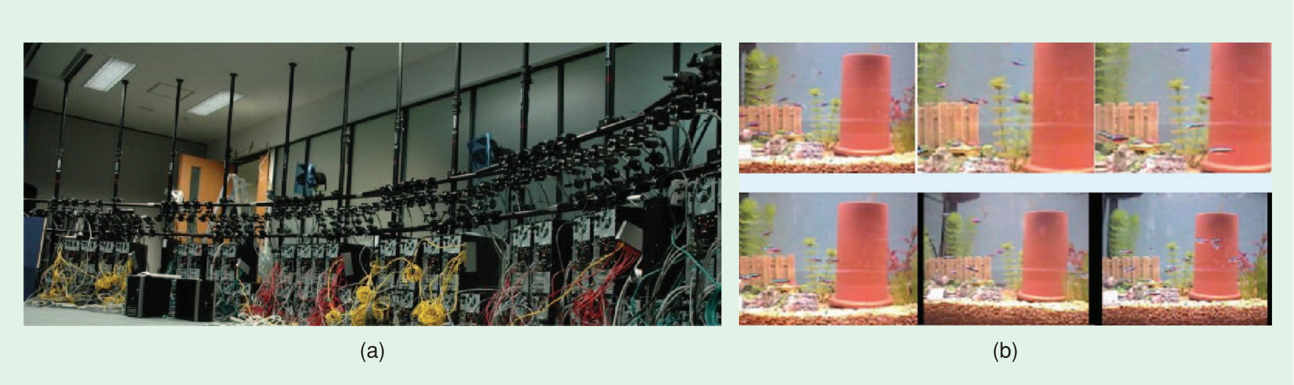
[FIG1] Basic components of a 3DTV/FTV system.

As seen in the movie *The Matrix*, successive switching of multiple real images captured at different angles can give the sensation of a flying viewpoint. In a similar way, Eye Vision [11] realized a flying virtual camera for a scene in a Super Bowl game. It used 33 cameras arranged around the stadium and controlled the camera directions mechanically to track the target scene. In these systems, however, no new virtual images are generated, and the movement of the viewpoint is limited to the predefined original camera positions; hence functionalities of 3DTV/FTV are not provided. It is in fact extremely challenging to realize these functionalities using a small number of sparsely positioned cameras in a large 3-D space such as a stadium.

To enable the use of 3DTV and FTV in real-world applications, the entire processing chain, including multiview image capture, 3-D scene representation, coding, transmission, rendering, and display, needs to be considered (Figure 1). There are numerous challenges in doing so. It is not easy to build a system that can capture and store a large number of videos in real time. An accurate calibration of camera position and color property is also required. From acquired multiview data, one should consider how to represent a 3-D scene that is more suitable for the latter processes. Depth reconstruction is one central task in 3-D representation but still a very difficult problem for rendering novel images precisely. The amount of multiview image data is usually huge, hence the data compressing and streaming with less degradation and delay over limited bandwidth are also challenging tasks.

In addition, there are also strong interrelations between all of the processes involved. The camera configuration (array or dome) and density (number of cameras) impose practical limitations on navigation and quality of rendered views at a certain virtual position. Therefore, there is a classical trade-off to consider between costs (for equipment, cameras, processors) and quality (navigation range, quality of virtual views). In general, the denser capturing of multiview images with a larger number of cameras provides a more precise 3-D representation, resulting in higher quality views through the rendering and display processes but requires a higher compression rate in the coding process, and vice versa. An interactive display that requires random access to 3-D data affects the performance of a coding scheme that is based on data prediction. Various types of quite diverse 3-D scene representations can be employed, which implies a number of different data types.

Here, let us briefly introduce one example of an FTV system [12]. This FTV system, implemented as a real-time complete processing chain, allows the user to freely control the viewpoint of a real dynamic 3-D scene. The developed multiview capturing system is shown in Figure 2(a), which consists of one host-server PC and 100 client PCs, each equipped with a high-definition camera (JAI PULNiX TM-1400CL). The interface between camera and PCs is created with Camera-Link. The host PC generates a synchronization signal and distributes it to all of the client PCs. This system is capable of capturing 100 synchronized high-resolution video signals at 30 fps. In addition, the camera positions can be easily changed. Figure 2(b) shows examples of generated free viewpoint images at various times and viewpoints. Complicated natural scenes including complex objects



[FIG2] A capturing system and generated free viewpoint images in FTV system [12].

such as small moving fishes, bubbles, and reflection of light from the aquarium glass are reproduced with high quality.

This special issue aims to provide a comprehensive overview and tutorial of the basic concepts and recent developments of 3DTV/FTV. We have selected eight articles in addition to this introductory article. Various challenging tasks involved in the capturing, representation and rendering processes and their state-of-the-art technical solutions are broadly reviewed in "Image-Based Rendering and Synthesis" [1]. The following article, titled "Plenoptic Manifolds" [2], provides an overview of image-based representation and introduces a new 3-D representation, *Plenoptic Manifolds*, based on analysis of structural coherence among multiview images. Model-based rendering approaches are reviewed in "High Quality Reconstruction from Multiview Video Streams" [3], where a sophisticated solution that can be applied to human actors for real-time rendering free-viewpoint video is described in detail. The next two articles focus on the compression process, "Compressing Time-Varying Visual Content" [4] broadly reviews coding techniques based on various types of 3-D representation, and "Multiview Compression" [5] provides a review of multiview video compression based on spatial and temporal similarity between multiview video. The streaming process is addressed in the article "3DTV over IP" [6], where the emphasis is placed on a tutorial overview of streaming of multiview video over the Internet Protocol (IP) for various 3-D display clients. Finally, the last two articles focus on 3-D display. A sampling problem in 3-D display technique based on ray-space representation is addressed in "Resampling, Antialiasing, and Compression in Multiview 3-D Displays" [7]. The other article, titled "3-D Displays and Signal Processing" [8], provides a comprehensive overview of various types of 3-D display.

3DTV/FTV is a state-of-the-art imaging system. In the past, image systems such as photography, film, and TV were distinct systems. At present, they are being digitized more and more, allowing them to be handled on the same platform as pixel-based systems. These pixel-based systems are undergoing rapid development toward increasing the number of pixels. Although super high-definition TV has about 100 times the number of pixels of standard-definition TV, still only one view is used. In the future, the demand for more pixels will level off and will be replaced with a demand for more views (i.e., more light rays in 3-D space). This is an evolution from pixel-based systems with a single image to ray-based systems with multiview images. The technologies for light ray capturing and display are making rapid progress and have created a huge opportunity for 3DTV/FTV to enter the consumer mass market in the near future. MVI will open the way to ray-based image engineering that provides breakthrough technologies to treat rays one by one.

In the following sections, this article provides the reader with a brief introduction to the sequence of fundamental processing steps required for 3DTV and FTV: 3-D scene representation, multiview image capturing and rendering, coding, and displaying of a 3-D scene. We hope that reading through these sections will aid the reader in understanding the articles in this issue.

3-D SCENE REPRESENTATION

The choice of a 3-D scene representation format is of central importance for the design of any 3DTV system. On one hand, the scene representation sets the requirements for multiview image capturing and processing. For instance using an image-based representation (see below) implies using a dense camera setting. A relatively sparse camera setting would only give poor rendering results of virtual views. Using a geometry-based representation (see below) in contrary implies the need for sophisticated and error prone image processing algorithms such as object segmentation and 3-D geometry reconstruction. On the other hand, the 3-D scene representation determines the rendering algorithms (and with that also navigation range, quality, etc.), interactivity, as well as compression and transmission if necessary.

In computer graphics literature, methods for 3-D scene representation are often classified as a continuum in between two extremes. The one extreme is represented by classical 3-D computer graphics. This approach can also be called geometry-based modeling. In most cases scene geometry is described on the basis of 3-D meshes. Real world objects are reproduced using geometric 3-D surfaces with an associated texture mapped onto them. More sophisticated attributes can be assigned as well. For instance, appearance properties (opacity, reflectance, specular lights, etc.) can significantly enhance the realism of the models.

Geometry-based modeling is used in applications such as games, Internet, TV, and movies. The achievable performance with these models might be excellent, typically if the scenes are purely computer generated. The available technology for both production and rendering has been highly optimized over the last few years, especially in the case of common 3-D mesh representations. In addition, state-of-the-art PC graphics cards are able to render highly complex scenes with an impressive quality in terms of refresh rate, levels of detail, spatial resolution, reproduction of motion, and accuracy of textures.

A drawback of this approach is that typically high costs and human assistance are required for content creation. Aiming at photorealism, 3-D scene and object modeling is often complex and time consuming, and it becomes even more complex if dynamically changing scenes are considered. Furthermore, an automatic 3-D object and scene reconstruction implies an estimation of camera geometry, depth structures, and 3-D shapes. With some likelihood, all these estimation processes generate errors in the geometric model. These errors then have an impact on the rendered images. Therefore, high-quality production of geometry model, e.g., for movies, is typically done user assisted.

The other extreme in 3-D scene representations is called image-based modeling and does not use any 3-D geometry at all. In this case virtual intermediate views are generated from available natural camera views by interpolation. The main advantage is a potentially high quality of virtual view synthesis avoiding any 3-D scene reconstruction. However, this benefit has to be paid by dense sampling of the real world with a sufficiently large number of natural camera view images. In general, the synthesis quality increases with the number of available views. Hence, typically a large number of cameras have to be set up to achieve high-performance rendering, and a tremendous amount of image data needs to be processed therefore. Contrarily, if the number of used cameras is too low, interpolation and occlusion artifacts will appear in the synthesized images, possibly affecting the quality.

Examples of image-based representations are ray space [13] or light field [14] and panoramic configurations including concentric and cylindrical mosaics [15]. All these methods do not make any use of geometry, but they either have to cope with an enormous complexity in terms of data acquisition or they execute simplifications restricting the level of interactivity.

In between the two extremes there exists a number of methods that make more or less use of both approaches and combine the advantages in some way. For instance, a Lumigraph [16] uses a similar representation as a light-field but adds a rough 3-D model. This provides information on the depth structure of the scene and therefore allows for reducing the number of necessary natural camera views. Other representations do not use explicit 3-D models but depth or disparity maps. Such maps assign a depth value to each sample of an image. Together with the original two-dimensional (2-D) image the depth map builds a 3-D-like representation, often called 2.5-D [17]. This can be extended to layered depth images [18], where multiple color and depth values are stored in consecutively ordered depth layers. A different extension is to use multiview video plus depth, where multiple depth maps are assigned to the multiple color images [19], [20]. Closer to the geometry-based end of the spectrum, methods are reported that use view-dependent geometry and/or view dependent texture [21]. Surface light fields combine the idea of light fields with an explicit 3-D model [22]. Instead of explicit 3-D mesh models also point-based representations or 3-D video fragments can be used [23].

More details can be found in survey papers [24], [25] and special issue articles [1], [2].

CAPTURING AND RENDERING

CAPTURING SYSTEM

STATIC SCENES

Capturing multiple views of a static scene is relatively simple because only a single camera is needed. One can move the camera along a certain predetermined path to take multiple images of the scene. Novel views can then be synthesized from the captured images, with or without the scene geometry. Note the camera position/geometry is assumed to be known, hence rendering with multiview imaging is not truly “geometry free.”

The camera geometry can be established in two ways. First, one can use a robotic arm, or a similar mechanism to control the movement of the camera. For instance, a camera gantry is used in [14] to capture light field, which assumes that the camera locations form a uniform grid on a 2-D plane. In concentric mosaics [15], a camera is mounted on the tip of a rotating arm, which captures a series of images whose centers of projection are along a circle. Turn table has been widely used in literature for capturing inner-looking images for the purpose of geometry reconstruction, which certainly falls into the general concept of multiview imaging. The second approach to obtain camera geometry is through calibration. In the work Lumigraph [16], the authors used a handheld camera to capture the scene. The scene contains three planar patterns, which are used for camera calibration. In [22], a camera attached to a spherical gantry arm is used to capture images roughly evenly over the sphere. Calibration is still performed to register the camera locations to the scene geometry obtained through range scan. When the scene itself contains a lot of interest points, it is possible to extract and match feature points directly for camera calibration (known as structure from motion in computer vision), such as the work in [26].

DYNAMIC SCENES

When the scene is dynamic, an array of cameras is needed. Most existing camera arrays contain a set of static cameras; hence the camera geometry can be pre-calibrated before scene capture. One exception is the self-reconfigurable camera array developed in [27], which has 48 cameras mounted on robotic servos. These cameras move during capturing to acquire better images for rendering. As a result, they have to be calibrated on-the-fly using a calibration pattern in the scene.

Capturing dynamic scenes with multiple cameras has a number of challenges. For instance, the cameras need to be synchronized if correspondence between images will be explored in the rendering stage. The amount of data captured by a camera array is often huge, and it is necessary to write these data into storage devices as fast as possible. Color calibration is another issue that needs to be addressed in order to render seamless synthetic views.

When the number of cameras in the array is small, synchronization between cameras is often simple. A series of 1394 FireWire cameras can be daisy chained to capture multiple videos, and the synchronization of exposure start of all the cameras are guaranteed on the same 1394 bus. Alternatively, the cameras’ exposure can be synchronized using a common external trigger. This is a very widely used configuration and can scale up to large camera arrays [20], [28]–[31]. In the worst case, where the cameras in the system cannot be genlocked [27], [32], camera synchronization can still be roughly achieved by pulling images from the cameras at a common pace from the computer. Slightly unsynchronized images may cause artifacts in scene geometry reconstruction for rapid-moving objects, but the rendering results may still be acceptable since human eyes are not very sensitive about details in moving objects.

When multiple videos are recorded simultaneously, the amount of data that needs to be stored/processed is huge. Most existing systems employ multiple computers to record and process the data from the cameras. The Stanford multicamera array [29] used a modular embedded design based on the IEEE1394 high speed serial bus, with an image sensor and MPEG2 compression at each node. Since video compression is performed on the fly, the system is capable of recording a synchronized video data set from over 100 cameras to a hard disk array using as few as one PC per 50 image sensors. The work in [32] introduced an interesting concept called distributed light field camera. Multiple computers are used to serve the data to the renderer upon request. Ideally, these computers can be integrated with the cameras, so that each camera can serve a few random light rays (pixels) when they are needed for rendering. This design minimizes the bandwidth required between the cameras and renderers, which is critical when hundreds of cameras are employed.

For low-cost camera arrays, it is difficult to guarantee that all cameras have the same color when capturing the same object. Color inconsistency across cameras may cause incorrect view-dependent color variation during rendering. The color calibration issue was rarely studied in literature. In [33], Joshi et al. proposed an iterative scheme to calibrate the sensors to match each other rather than a common standard. This yields better color consistency between cameras, which is more suitable for multiview imaging applications.

SCENE GEOMETRY

The geometry of the scene is usually very useful during the rendering of 3DTV or FTV. In this section we briefly review a few mechanisms that can be used to obtain the scene geometry directly instead of deriving from images.

One well-known depth discovery technique is based on triangulation. A laser stripe is scanned across the scene, which is captured by a camera positioned at a distance from the laser pointer. The range of the scene is then determined by the focal length of the camera, the distance between the camera and the laser pointer, and the observed stripe position in the captured image. Due to the limited sweeping speed of the stripe lights, this method is often used to capture the geometry of static scenes. Recent improvements in the triangulation method can use multiple stripe patterns to recover scene depth with one single image. However, its application in 3DTV or FTV is still limited because the stripe pattern may change the scene color/texture when images are simultaneously captured for rendering.

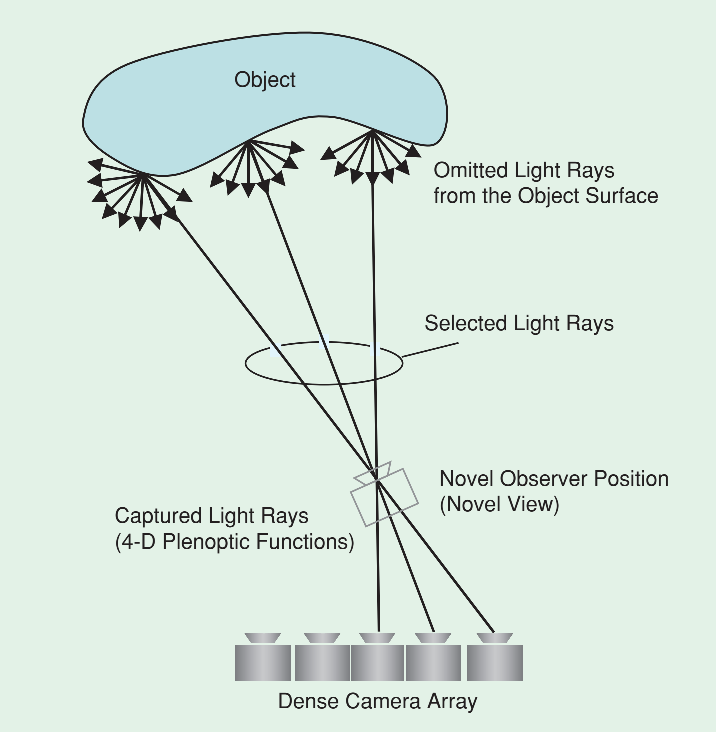
For dynamic scenes, most commercially available 3-D cameras are based on the time-of-flight principle. Laser beams (often in the infrared spectrum) are emitted to the scene, and the reflections are collected by the device to measure the time of flight. Broadly speaking, time-of-flight range sensors can be divided into two main categories: pulsed wave and continuous modulated wave. Pulsed-wave sensors measure the time of delay directly, while continuous modulated-wave sensors measure the phase shift between the emitted and received laser beams to determine the scene depth. One example of pulsed wave sensors is the 3-D terrestrial laser scanner systems manufactured by Riegl (http://www.riegl.com/). Continuous modulated wave sensors include SwissRanger (http://www.swissranger.ch/) and ZCam Depth Camera from 3DV Systems (http://www.3dvsystems.com/).

IMAGE-BASED RENDERING

LIGHT FIELD REPRESENTATION AND RENDERING

Imagine that it is possible to record all light rays traveling from an object's surface to arbitrary observer positions. In this case, we could create correct novel views from various perspectives simply by selecting the necessary light rays from the recorded rays. This is obvious; however this concept has brought us a new paradigm, called image-based rendering (IBR), and IBR research has attracted much attention since the early 1990s. The main reason is that it allows photo-realistic rendering with a much lighter computation load, independent of the scene complexity. The history of this concept is briefly described in [34].

The plenoptic function [35] was introduced to describe these light rays in seven dimensions (7-D), using the parameters of 3-D position, 2-D direction, wave-length (color) and time. Consider the case of fixed color component and time, it reduces to a five-dimensional function. Furthermore, since it can be assumed that light intensity remains constant along its trajectory, an arbitrary light ray can be described with a 4-D plenoptic function. This representation of light rays is called ray space [13] or light field [14], which is the most useful and practical representation. The light field is usually acquired with a 2-D planer array of cameras (Figure 3). In this case, each light ray can be parameterized by the 2-D camera position and the 2-D pixel position. If we have light field data densely sampled on a plane, we can generate a novel view correctly by selecting (resampling) the necessary light rays. Various dimensionally reduced versions of the plenoptic function have been surveyed in [24], [25], and [2].



[FIG3] **Light field capturing and rendering.**

We can consider this capturing problem in IBR as a sampling problem of the plenoptic function for the object. The goal is to sample the plenoptic function densely enough to reconstruct the original continuous plenoptic function through resampling the sampled function. Sampling theory provides us with an answer as to how densely we need to capture the light field. Shum et al. [36] applied this theory to the light field and theoretically derived the minimum camera spacing density and the interpolation filter. Although it has been possible to build a camera array system with more than 100 cameras, it is unfortunately impossible to space cameras densely enough for most practical applications, due to camera size. Instead of using multiple cameras, it is also possible to acquire a much denser light field through the use of an array of micro lenses and a single camera; however the range of viewing positions and directions is much more limited. It is possible to acquire light fields densely and correctly for static scenes by moving a hand-held video camera, however the camera position must be precisely obtained.

VIEW INTERPOLATION FROM SPARSELY SAMPLED LIGHT FIELD

As mentioned above, equipment is not yet developed for practical applications that can acquire light fields or the plenoptic function with the required density. For novel view generation using light fields sampled with insufficient density, we need to interpolate the missing light rays. The intensity of the desired missing ray is computed by blending a small number (typically four) of acquired rays. Here, we take the average (i.e., the mean value) of the acquired rays in the blending process. The problem is that of which rays should be used for this average. If the depth of the missing ray is known, then we can easily select and use the rays corresponding to this depth. Hence, we need to estimate the depth beforehand.

One of the most widely used and effective process of depth estimation in an image-based approach is color consistency among rays. Figure 4 shows the basic concept of depth estimation using color consistency, called the plane-sweeping algorithm [37]. Multiple hypothetical depths are set in the scene; let the number of depths be N and each depth be $d_i (i = 1, \dots, N)$. The corresponding light rays at the intersection point of the missing rays and the hypothetical plane at d_i are selected, and the mean (μ_i) and variance (σ_i) of their color values are computed. The simplest definition of the color consistency is the inverse of the variance. The estimated depth chosen is the depth that gives the highest color consistency (i.e., the minimum variance). Based on this estimated depth, the color intensity of the missing ray, L, is computed as

$$L = \mu_{\arg\min\{\sigma_i\}}. \tag{1}$$

The advantage of this plane-sweeping approach is that it is very simple and does not need to explicitly solve a computationally expensive feature-correspondence problem. Problems arise however in correctly estimating the depth for textureless regions, and an incorrect depth creates visible artifacts such as noise.

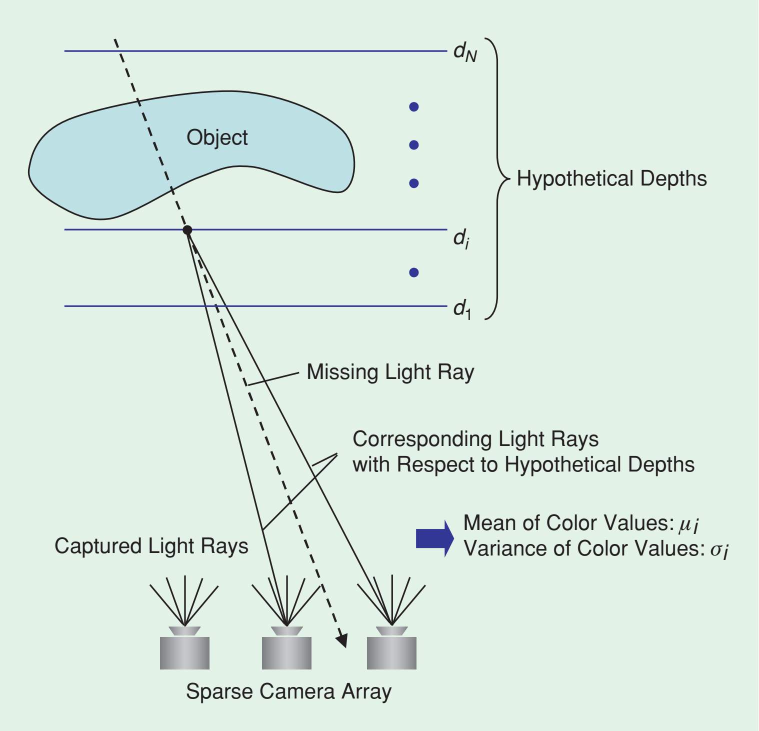
Our goal is not necessarily to estimate the depth but to generate the missing ray. Instead of identifying the depth uniquely, a novel concept of estimating the probability of each hypothetical depth being the actual depth was proposed in [38]. This effectively handles the depth ambiguity of textureless regions. The simplest approach to doing this is to assume that the probability P_i of d_i being the actual depth is inversely proportional to the color variation σ_i (i.e., proportional to the color consistency), which we define as

$$P_i = \frac{1/\sigma_i}{\sum_{i=1}^{N}(1/\sigma_i)}. \tag{2}$$

It can be intuitively understood that P_i has a peak at the actual depth for textured regions, which means the depth can be estimated correctly to be the depth that gives the highest probability; for textureless regions, P_i has almost constant values over the depths and hence the depth cannot be determined uniquely. The color value L of the missing rays is computed as a weighted average of the mean color values with the corresponding probability

$$L = \sum_{i=1}^{N} P_i \mu_i. \tag{3}$$

This works better even for the textureless regions and noisy light field data than the conventional depth estimation based method.


d_N
Object
Hypothetical Depths
d_i
d_1
Missing Light Ray
Corresponding Light Rays with Respect to Hypothetical Depths
Mean of Color Values: μ_i
Variance of Color Values: σ_i
Captured Light Rays
Sparse Camera Array

[FIG4] Interpolation of missing light ray from sparsely sampled light field.

The above method, using P_i as a coefficient, fuses color mean values μ_i into the missing color value L. Since P_i depends upon the scene and needs to be computed for every frame for dynamic scenes. A different fusion method was presented in [39] that does not need to estimate any scene information, although it requires a denser camera arrangement than the depth-probability method above. Assuming texture s_i exists on each hypothetical plane, this method models the missing color value L as the sum of the textures

$$L = \sum_{i=1}^{N} s_i. \tag{4}$$

The mean color value μ_i is modeled as a linear combination of the assumed textures with blending artifacts:

$$\begin{cases} \mu_1 &= s_1 + h_{12}(s_2) + h_{13}(s_3) \\ & \quad + \cdots + h_{1N}(s_N) \\ \mu_2 &= h_{21}(s_1) + s_2 + h_{23}(s_3) \\ & \quad + \cdots + h_{2N}(s_N) \\ & \vdots \\ \mu_N &= h_{N1}(s_1) + \cdots \\ & \quad + h_{NN-1}(s_{N-1}) + s_N. \end{cases} \tag{5}$$

The function $h_{ij}(\cdot)$ denotes a point spread function (PSF) that causes blending artifacts on the jth textures $s_j (j = 1, 2, \ldots, N)$. All PSFs are spatially varying filters but can be simply determined in such a way that we assume a point light source at depth d_j and compute the mean value μ_i on different depth d_i. The resulting mean value is the filter $h_{ij}(\cdot)$. This computation can be done beforehand, independent of the scene. This simultaneous equation for the textures is iteratively solved using the Gauss-Seidel method starting from arbitrary initial solutions. Solving this simultaneous equation is ill conditioned and scene textures cannot be obtained correctly; the obtained each texture contains blurry textures that are supposed to exist at other depths. The sum of these solutions however provides a good approximation to the missing ray L.

These image-fusion approaches are effective for scenes that consist of a nonoccluded surface possessing the Lambertian property (reflected light intensity is constant for every direction). Various IBR approaches for general scenes is reviewed in [1].

MODEL-BASED RENDERING

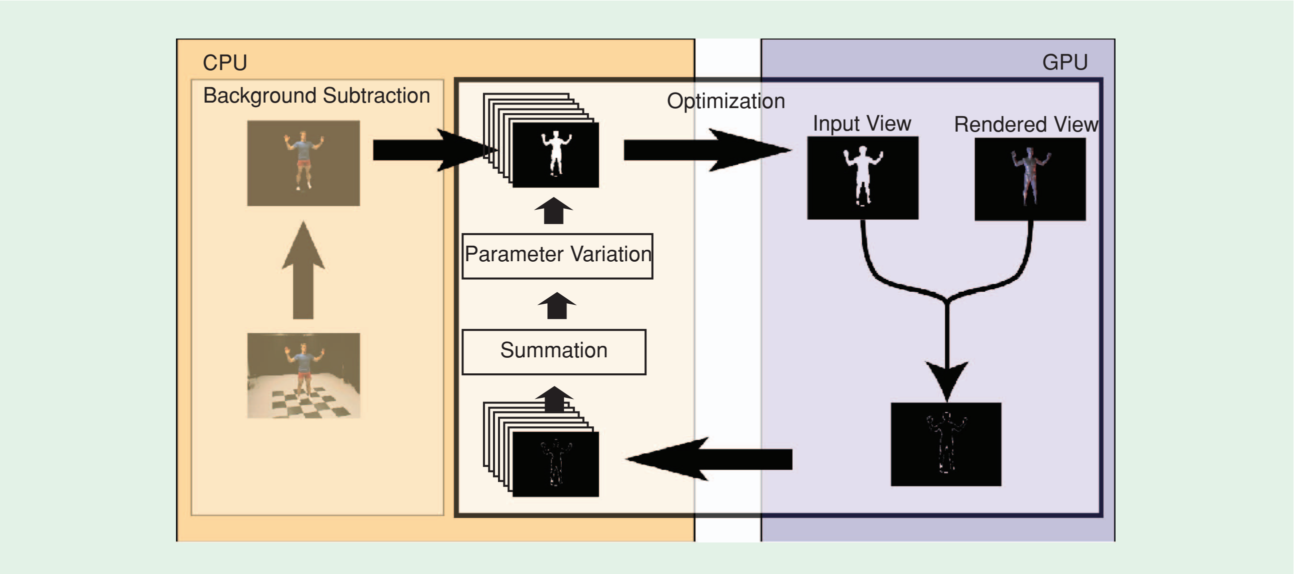
MODEL-BASED FTV ACQUISITION

In many FTV scenarios, the object that is being recorded is known in advance. Suitably implemented, such a priori knowledge can be used to bias the scene reconstruction outcome towards plausible results only. Of course, a suitable model of the recorded object(s) must be available. A model also enables enforcing low-level as well as high-level constraints about the object's motion, from temporally coherent movement to anatomy-consistent motion. Another advantage of model-based FTV is that a priori model geometry can be highly detailed, which facilitates high-quality rendering results and circumvents rendering inaccuracies due to poorly resolved geometry. In general, model-based FTV methods are able to achieve more robust and authentic rendering results than methods ignorant of recorded scene content.

While for motion capture purposes, it is sufficient to recover model animation parameters, FTV imposes the additional demand that the resulting model must be able to produce convincing rendering results. The challenge in model-based FTV therefore is how to automatically, robustly, and visually consistently match a parameterized 3-D geometry model to recorded image content.

One model-based FTV method that is suitable for synchronized multiview video footage consists of matching model to object silhouettes based on an analysis-by-synthesis approach [28], as shown in Figure 5. The object's silhouettes, as seen from the different camera viewpoints, are used to match the model to the recorded video images: The foreground in all video images is segmented and binarized. At the same time, the 3-D object model is rendered from all camera viewpoints using conventional graphic hardware, after which the rendered images are thresholded to yield binary masks of the model's silhouettes. Then, the rendered model silhouettes are compared to the corresponding image silhouettes: as comparison measure, or matching score, the number of silhouette pixels is used that do not overlap when putting the rendered silhouette on top of the recorded silhouette. Conveniently, the logical exclusive-or (XOR) operation between the rendered image and the recorded image yields those silhouette pixels that are not overlapping. By summing over the nonoverlapping pixels for all images, the matching score is obtained. This matching score can be evaluated very efficiently on contemporary graphics hardware. To adapt model parameter values such that the matching score becomes minimal, a standard numerical nonlinear optimization algorithm, e.g. the Powell optimization method [40], runs on the CPU. For each new set of model parameter values, the optimization routine evokes the matching score evaluation routine on the graphics card which can be evaluated many hundred times per second. After convergence, object texture can be additionally exploited for pose refinement [41].

One advantage of model-based analysis is the low-dimensional parameter space when compared to general reconstruction methods (Figure 6): The parameterized 3-D model may provide only a few dozen degrees of freedom that need to be determined, which greatly reduces the number of potential local minima. Many high-level constraints are already implicitly incorporated into the model, such as kinematic capabilities. Additional constraints can be easily enforced by making sure that all parameter values stay within their anatomically plausible range during optimization. Finally, temporal coherence is straightforwardly maintained by allowing only some maximal rate of change in parameter value from one time step to the next.

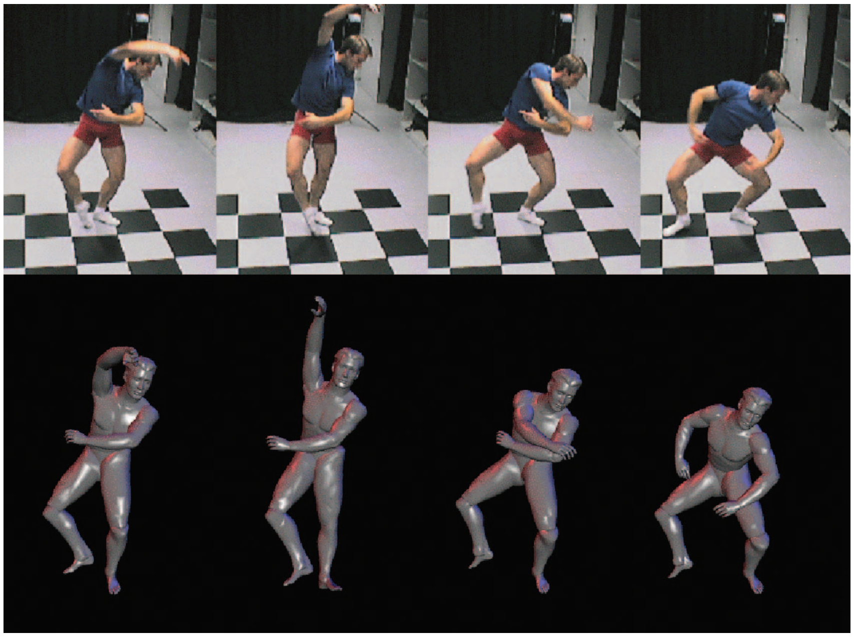


[FIG5] Analysis-by-synthesis: To match the geometry model to the multiview video footage, the foreground object is segmented and binarized, and the 3-D model is rendered from all camera viewpoints. The boolean XOR operation is executed between the reference images and the corresponding model renderings. The number of nonoverlapping pixels serves as matching score. VIe numerical optimization, model parameter values are varied until the matching score is minimal.

RENDERING FROM ARTICULATED 3-D GEOMETRY MODELS

After model-based motion capture, a high-quality 3-D geometry model is available that closely, but not exactly, matches the dynamic object in the scene. For photorealistic rendering results, the original video footage must be applied as texture to the model. By making efficient use of multivideo footage, time-varying cloth folds and creases, shadows, and facial expressions can be faithfully reproduced to lend a very natural, dynamic appearance to the rendered object. Projective texture mapping is a well-known technique to apply images as texture to triangle-mesh models. To achieve optimal rendering quality, however, it is necessary to process the video textures offline prior to real-time rendering [28]: local visibility must be considered correctly to avoid any rendering artifacts due to the inevitable small differences between model geometry and the true 3-D object surface. Also, the video images, which are taken from different viewpoints, must be blended appropriately to achieve the impression of one consistent object surface texture.

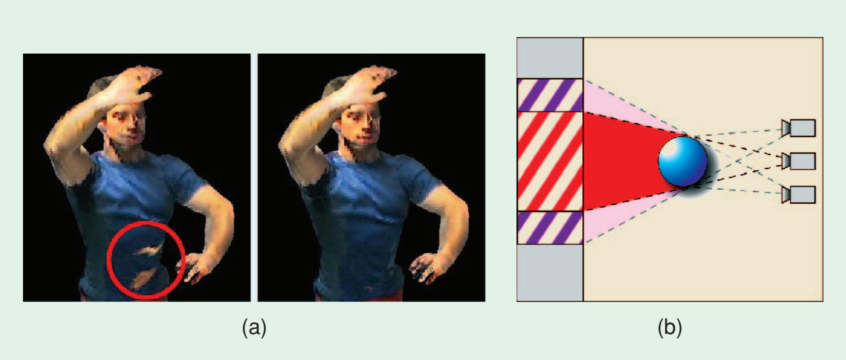
Because model geometry is not exact, the reference image silhouettes do not correspond exactly to rendered model silhouettes. When projecting the reference images onto the model, texture belonging to some frontal body segment potentially leaks onto other segments farther back [Figure 7(a)]. To avoid such artifacts, each reference view's *penumbral region* must be excluded during texturing. To determine the penumbral region of a camera, vertices of zero visibility are determined not only from the camera's actual position but also from a few slightly displaced virtual camera positions [Figure 7(b)]. For each reference view, each vertex is checked whether it is visible from all camera positions, actual as well as virtual. A triangle is projectively textured using a reference image only if all of its three vertices are completely visible from that camera.



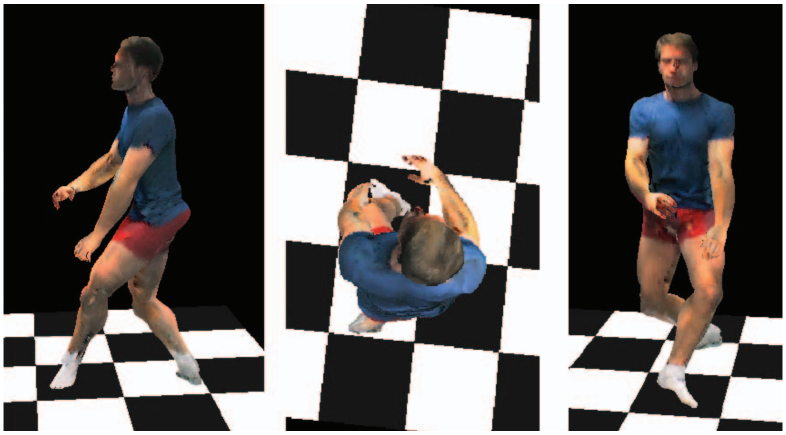
[FIG6] From only eight video cameras spaced all around the scene, model-based FTV can fully automatically capture the complex motion of a jazz dancer.

Most surface areas of the model are seen from more than one camera. If the model geometry corresponded exactly to that of the recorded object, all camera views could be weighted according to their proximity to the desired viewing direction and blended without loss of detail. However, model geometry has been adapted to the recorded person by optimizing only a comparatively small number of free parameters. The model is also composed of rigid body elements which is clearly an approximation whose validity varies, e.g., with the person's apparel. In summary, the available model surface can be expected to locally deviate from true object geometry. Accordingly, projectively texturing the model by simply blending multiple reference images causes blurred rendering results, and model texture varies discontinuously when the viewpoint is moving. Instead, by taking into account triangle orientation with respect to camera direction, high-quality rendering results can still be obtained for predominantly diffuse surfaces [28].

After uploading the 3-D model mesh and video cameras' projection matrices to the graphics card, the animated model is ready to be interactively rendered. During rendering, the multiview imagery, predetermined model pose parameter values, visibility information, and blending coefficients must be continuously uploaded, while the view-dependent texture weights are computed on the fly on the GPU. Easily achieving real-time rendering frame rates, views of the object from arbitrary perspective are possible, as well as freeze-and-rotate shots, fly-around sequences, close-ups, slow motion, fast forward, or reverse play (Figure 8). More detail can be given in [3] in this issue.



[FIG7] Penumbral region determination: (a) Small differences between object silhouette and model outline can cause texture of frontal model segments to leak onto segments farther back. (b) By projecting each reference image onto the model also from slightly displaced camera positions, regions of dubious visibility are determined. These are excluded from texturing by the respective reference image. *((a) reprinted from [28] ©2003 ACM).*



[FIG8] Model-based FTV: The user can freely move around the dynamic object at real-time rendering frame rates.

CODING

Three-dimensional scene representation formats integrate various types of data, such as multiview video, and geometry data in form of depth or 3-D meshes. In general, these results in a tremendous amount of data that needs to be transmitted or stored. Therefore, efficient compression is a key condition for the success of such applications. Further, the availability of open international standards is in general an important enabling factor for the development of markets in the media business. ISO/IEC JTC 1/SC 29/WG 11 (Moving Picture Experts Group—MPEG) is one of the international standardization bodies that play an important role in digital media standardization [42].

The compression of 3-D data has recently received a lot of attention in research and development. Technology has reached a good level of maturation, however, since the field is still very young compared for instance to classical 2-D video coding, there is still a lot of room for improvement and optimization. The area of 3-D compression may be categorized into 3-D meshes and pixel-type data, such as multiview images, video and depth or disparity. Associated data such as external and internal camera parameters defining the 3-D space and 2-D–3-D relations need to be considered in addition.

Three-dimensional mesh compression can be divided into static and dynamic. Static 3-D mesh compression has been studied extensively. Efficient algorithms are available; some providing extended functionality such as progressive decoding. Dynamic 3-D meshes describe moving objects and change over time. This is more interesting for multiview video and 3DTV applications as studied in this special issue. This research area has gained significant attention recently and various algorithms have been proposed. More details can be found in [43] and [4].

Multiview video coding (MVC) has also gained significant attention recently. Numerous papers have been published and significant progress has been made. Multiview video shows the same 3-D scene from different viewpoint, resulting in a tremendous amount of raw video data. However, the different camera signals contain a large amount of statistical dependencies. The key for efficient MVC lies in exploitation of these interview redundancies in addition to temporal redundancies [44]. An example for such a prediction structure is shown in Figure 9. Results indicate that such specific MVC may provide the same video quality at half the bitrate compared to independent encoding of all camera signals. More details can be found in [43], [4], and [5].

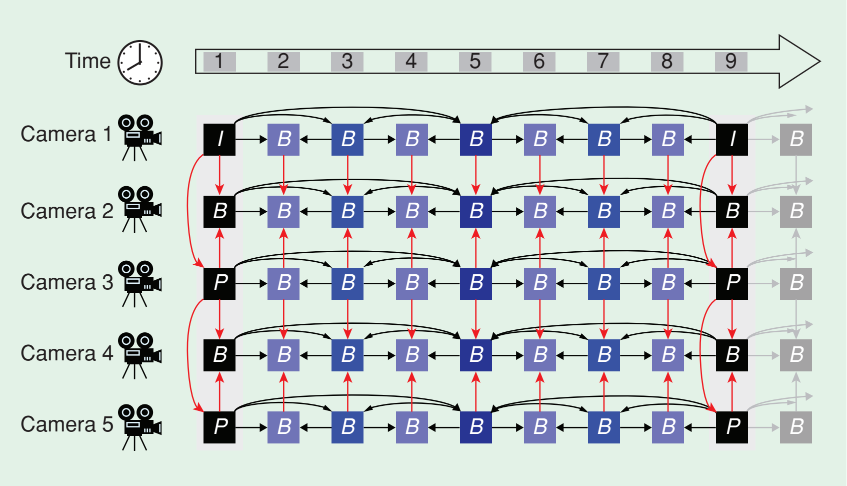
As mentioned above 3-D geometry can also be represented by per-pixel depth data associated with the color video as illustrated in Figure 10. Depth needs to be clipped in between 2 extremes Z_{near} and Z_{far} and the range in between is most often scaled nonlinearly. Quantization with 8 b is often sufficient (however, e.g. not for large depth outdoor scenes) resulting in a grayscale image as shown in Figure 10. Three-dimensional points close to the camera have large Z values and distant points have small Z values in this formulation. Then, consecutive depth maps can be regarded as a monochromatic video signal and encoded with any available video coded. Investigations have shown that such data can be encoded very efficiently, e.g. at 5–10% of the bit rate that is needed to encode the associated color video at a good quality [17]. This means that the extension from 2-D video to 3-D video comes at a very limited overhead. However, this is only true for a limited navigation range, i.e., rendered virtual views are close to the original camera position. In the case of extended navigation, artifacts of depth compression can lead to very annoying artifacts in rendered views especially along physical object border that result in depth discontinuities [45]. Therefore, efficient edge preserving compression of depth data needs to be investigated in the future. More details can be found in [43], [4], and [5].

In conclusion, various 3-D scene representation formats enabling different types of 3DTV and multiview video systems and applications are available and under further study. Efficient compression algorithms for the different types of data involved are available; however, there is still room for further research. MPEG is continuously working on developing and updating standards for these representations along with associated compression to provide the basis for development of mass markets in media business. For instance, a dedicated standard for MVC (extension of H.264/AVC, i.e., MPEG-4 Part 10 Amendment 4) is currently under development and scheduled to be available in early 2008 [46].

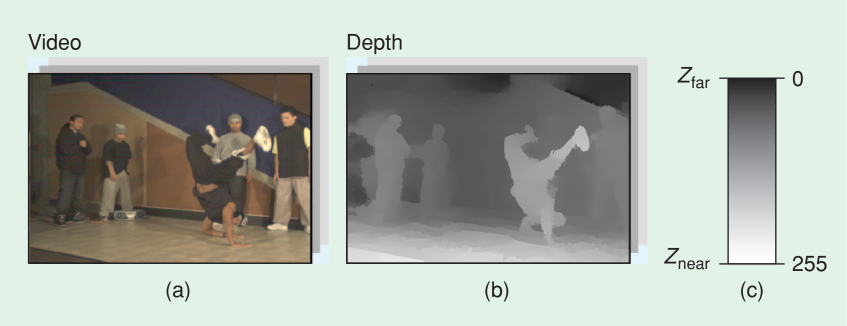
3-D DISPLAY

As explained earlier, 3DTV implies depth perception of the observed scenery instead of a flat 2-D image. This has been investigated over decades. A conventional stereo video system exploits the human visual perception to create the depth impression. The basic principle is to present two images of the observed scenery to the user that correspond to the two views of the eyes. From this, the human brain generates the depth perception. The technical challenge is to display these two images while ensuring that each eye only sees one of them. A well-known approach is to overlay two differently colored images in an anaglyph representation and to use glasses with color filters (e.g., red-green or red-cyan glasses). However, the visual quality of such systems is quite limited due to unnatural colors. More advanced systems use temporal interleaving of the left and right eye view with synchronized shutter glasses or polarization projection with polarizing glasses.

Such 3-D display systems have gained significance in niche markets and have already become practical for various applications such as 3-D cinemas (e.g., IMAX theatres), PC gaming, professional and scientific visualization, etc., raising user awareness of 3-D technologies as well as content creation. Important content providers see a market for 3-D movies; existing 2-D material is being converted to 3-D. Hollywood animation movies are released in 2-D and 3-D simultaneously already. Some program production companies in Japan regularly create stereo video content for business purposes. Three-dimensional clips and images are populating the internet. Basically every 3-D computer game can be enjoyed with 3-D



[FIG9] Multiview video coding structure combining inter-view and temporal prediction.



[FIG10] Video plus depth data representation format consisting of regular 2-D color video and accompanying 8-bit depth-images.

perception with very limited hardware costs to upgrade any common end-user PC.

However, conventional stereo video systems require wearing specific glasses, which is among other reasons considered to be the main obstacle for development of wide 3DTV user markets. A living room environment requires new concepts for 3-D displays, for watching TV with family and friends. So-called auto-stereoscopic displays may overcome this situation in the very near future. Such displays provide 3-D depth perception without the necessity of wearing glasses. Technology is based on lenticular screens or parallax barrier systems and is already quite advanced. Displays are on the market, so far mainly for professional customers, but some manufacturers plan introducing them to broad end-user markets in the near future. Backward compatible concepts for introduction of 3DTV via broadcast or DVD are developed.

The multiview display principle has been used to construct auto-stereoscopic displays. With this approach, parallax images, which are perspective projections of three-dimensional objects, are displayed to converge to the corresponding view points. Signal processing for multiview 3-D displays is reviewed in the special issue paper [7].

More recently, the directional display technique has come to be used. With this technique, directional images, which are orthographic projections of objects, are displayed with nearly parallel rays [47]. The directional display technique provides more natural motion parallax than the multiview technique. The 3-D impression becomes free from the visual fatigue effect caused by the accommodation-convergence conflict.

However, these systems still rely on a fake of the human visual system. Real 3-D images can be generated by even more advanced approaches such as volumetric or holographic displays, which will most probably provide us with real 3-D sensation in the mid term future. So far such systems are under development and not yet mature enough. A detailed overview of 3-D display technology can be found in [8].

ACKNOWLEDGEMENT

This issue attracted much attention from many researchers in various fields, and we received more than 60 paper proposals. Due to limited schedule and allocated pages for such a large number of submissions, we had to select papers within a short time. We would like to thank all of authors who submitted their proposal to this special issue and to express many thanks to all the reviewers as well. We also express thanks to Prof. Douglas B Williams, Special Sections area editor, for his great support and help. We would further like to thank the Interactive Visual Media Group of Microsoft Research for providing the Breakdancers data sets used in Figure 10.

GUEST EDITORS

Akira Kubota received the B.E. degree in electrical engineering from Oita University, Japan, in 1997, the M.E. and Dr.E. degrees in electrical engineering from the University of Tokyo, Japan, in 1999 and 2002, respectively. He is an assistant professor at the Department of Information Processing at Tokyo Institute of Technology, Yokohama, Japan. From September 2003 to July 2004, he was with the Advanced Multimedia Processing Laboratory at Carnegie Mellon University as a research associate. His research interests include image-based rendering and visual reconstruction.

Aljoscha Smolic received the Dr.-Ing. degree from Aachen University of Technology in 2001. Since 1994 he has been with Fraunhofer HHI, Berlin, Germany. He conducted research in various fields of video processing, video coding, computer vision and computer graphics. He is an adjunct professor at the Technical University of Berlin. He is an area editor for *Signal Processing: Image Communication* and guest editor for *IEEE Transactions on Circuits and Systems for Video Technology* and *IEEE Signal Processing Magazine*. He chaired the MPEG ad hoc group on 3DAV pioneering standards for 3-D video. Currently he is editor of the Multiview Video Coding (MVC) standard.

Marcus Magnor heads the Computer Graphics Lab of the computer science department at the Technical University Braunschweig. He received his B.A. (1995) and M.S. (1997) in physics from the Wurzburg University and the University of New Mexico, respectively, and his Ph.D. (2000) in electrical engineering from Erlangen University. His research interests include the complete visual information processing pipeline, from image formation, acquisition, and analysis to view synthesis, perception, and cognition.

Masayuki Tanimoto received B.E., M.E. and Dr.E. degrees in electronic engineering from the University of Tokyo in 1970, 1972, and 1976, respectively. Since 1991, he has been a professor at the Graduate School of Engineering, Nagoya University. He was vice president of ITE, chairperson of Technical Group on Communication Systems of IEICE, chairperson of the steering committee of Picture Coding Symposium of Japan, IEICE councilor, ITE councilor, and Tokai Section chair of IEICE. He received IEICE Fellow Award, IEICE Achievement Award, ITE Fellow Award and Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology.

Tsuhan Chen received the B.S. degree in electrical engineering from the National Taiwan University in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, California, in 1990 and 1993, respectively. He has been with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, since October 1997, where he is currently a professor. He directs the Advanced Multimedia Processing Laboratory. He helped create the Technical Committee on Multimedia Signal Processing, as the founding chair, and the Multimedia Signal Processing Workshop. His endeavor later evolved into founding of the IEEE Transactions on Multimedia and the IEEE International Conference on Multimedia and Expo. He was editor-in-chief for *IEEE Transactions on Multimedia* for 2002–2004. He is a Fellow of the IEEE.

AUTHOR

Cha Zhang received his B.S. and M.S. degrees from Tsinghua University, Beijing, China in 1998 and 2000, respectively, both in electronic engineering, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, in 2004. Since 2004, he has been a researcher with Microsoft Research. His current research focuses on automated lecture rooms, audio/video processing for multimedia collaboration, computer vision, etc. He has also worked on image-based rendering (compression/sampling/rendering), 3-D model retrieval, active learning, and peer-to-peer networking.

REFERENCES

[1] S.C. Chan, H.Y. Shum, and K.T. Ng, "Image-based rendering and synthesis," *IEEE Signal Processing Mag*. vol. 24, no. 7, pp. 22–33, Nov. 2007.

[2] J. Berent and P. Luigi Dragotti, "Plenoptic manifolds," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 34–44, Nov. 2007.

[3] C. Theobalt, N. Ahmed, G. Ziegler, and H.-P. Seidel, "High-quality reconstruction from multiview video streams," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 45–57, Nov. 2007.

[4] K. Müller, P. Merkle, and T. Wiegand, "Compressing time-varying visual content," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 58–67, Nov. 2007.

[5] M. Flierl and B. Girod, "Multiview video compression," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 66–76, Nov. 2007.

[6] A. Murat Tekalp, E. Kurutepe, and M. Reha Civanlar, "3DTV over IP," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 77–87, Nov. 2007.

[7] M. Zwicker, A. Vetro, S. Yea, W. Matusik, H. Pfister "Resampling, antialiasing, and compression in multiview 3-D displays," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 88–96, Nov. 2007.

[8] J. Konrad and M. Halle, "3-D displays and signal processing," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 97–111, Nov. 2007.

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

[10] M. Tanimoto, "Free viewpoint television—FTV," in *Proc. Picture Coding Symp. 2004*, Dec. 2004.

[11] CBS Broadcasting Inc., http://www.cbs.com/.

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

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

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

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

[16] S.J. Gortler, R. Grzesczuk, R. Szeliski, and M.F. Cohen, "The Lumigraph," in *Proc. ACM SIGGRAPH'96*, Aug. 1996, pp. 43–54.

[17] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. Ijsselsteijn, M. Pollefeys, L. Vangool, E. Ofek, and I. Sexton, "An evolutionary and optimised approach on 3D-TV," *IBC 2002, Int. Broadcast Convention*, Amsterdam, Netherlands, Sept. 2002.

[18] J. Shade, S. Gortler, L.W. He, and R. Szeliski, "Layered Depth Images," in *Proc. SIGGRAPH'98*, Orlando, FL, July 1998.

[19] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process. Image Commun.*, Special Issue on 3DTV, Feb. 2007.

[20] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. ACM SIGGRAPH and ACM Trans. Graphics*, Los Angeles, CA, Aug. 2004.

[21] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. SIGGRAPH 1996*, 1996, pp. 11–20.

[22] D. Wood, D. Azuma, W. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle, "Surface Light Fields for 3D Photography," in *Proc. SIGGRAPH* 2000.

[23] S. Wurmlin, E. Lamboray, and M. Gross, "3D video fragments: Dynamic point samples for real-time free-viewpoint video," *Comput. Graph.,* (Special issue on coding, compression and streaming techniques for 3D and multimedia data), vol. 28, no. 1, pp. 3–14, 2004.

[24] H.-Y. Shum, S.B. He, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, 2003.

[25] C. Zhang and T. Chen, "A survey on image-based rendering—Representation, sampling and compression," *EURASIP Signal Process. Image Commun.*, vol. 19, pp. 1–28, Jan. 2004.

[26] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, "Visual modeling with a hand-held camera," *Int. J. Comput. Vis.*, vol. 59, no. 3, pp. 207–232, 2004.

[27] C. Zhang and T. Chen, "A self-reconfigurable camera array," *Eurograph. Symp. Rendering 2004*, Norrkoping, Sweden, Jun. 2004.

[28] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," in *Proc. ACM Conf. Comput. Graph. (SIGGRAPH'03)*, 2003, pp. 569–577.

[29] B. Wilburn, M. Smulski, H.-H. K. Lee, M. Horowitz, "The light field video camera," in *Proc. Media Processors 2002, SPIE Electronic Imaging,* 2002.

[30] T. Kanade, H. Saito, S. Vedula, "The 3D room: Digitizing time-varying 3D events by synchronized multiple video streams," Tech. Rep. CMU-RITR-98-34, 1998.

[31] 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," *IEEE 2006 Int. Conf. Multimedia & Expo*, July 2006, pp. 437–440.

[32] J.C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera," in *Proc. Eurograph. Workshop Rendering 2002*, (2002), pp. 1–10.

[33] N. Joshi, B. Wilburn, V. Vaish, M. Levoy, and M. Horowitz, "Automatic color calibration for large camera arrays," UCSD CSE Tech. Rep. CS2005-0821, May 2005.

[34] M. Levoy, "Light fields and computational imaging," *IEEE Computer*, vol. 39, no. 8, pp. 46–55, Aug., 2006.

[35] E.H. Adelson, J.R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. Cambridge, MA: MIT Press, 1991.

[36] J.-X. Chai, X. Tong, S.-C. Chany, H.-Y. Shum, "Plenoptic sampling," in *Proc. SIGGRAPH 2000*, 2000, pp. 307–318.

[37] R.T. Collins, "Space-sweep approach to true multi-image matching," in *Proc. of CVPR'96*, pp. 358–363, 1996.

[38] Y. Kunita, M. Ueno, and K. Tanaka, "Layered probability maps: Basic framework and prototype system," in *Proc. of the ACM symposium on Virtual reality software and technology (VRST'06)*, pp. 181–188, 2006.

[39] A. Kubota, K. Takahashi, K. Aizawa, T. Chen, "All-focused light field rendering," in *Proc. of Eurographics Symposium on Rendering (EGSR2004)*, pp. 235–242, 2004.

[40] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, Cambridge Univ. Press, ISBN 0521431085, 1992.

[41] C. Theobalt, J. Carranza, M. Magnor, J. Lang, and H.-P. Seidel, "Combining 3D flow fields with silhouette-based human motion capture for immersive video," *Graphical Models* (Special issue pacific graph. '03), vol. 66, no. 6, pp. 333–351, 2004.

[42] A. Smolic, K. Müller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video—Technologies, applications and MPEG standards," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, Ontario, Canada, July 2006.

[43] A. Smolic, K. Müeller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G. Triantafyllidis, A. Koz, "Coding algorithms for 3DTV—A survey," *IEEE Trans. Circuits Syst. Video Technol.,* (Special Issue 3DTV MVC), Oct. 2007.

[44] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.,* (Special Issue 3DTV MVC), Oct. 2007.

[45] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. EEE Int. Conf. Image Processing 2007*, Oct. 2007.

[46] A. Vetro, Y. Su, H. Kimata, and A. Smolic, "Joint draft 2.0 on multi-view video coding," *Joint Video Team*, Doc. JVT-V209, Marrakech, Morocco, Jan. 2007.

[47] T. Saishu, S. Numazaki, K. Taira, R. Fukushima, A. Morishita, Y. Hirayama, "Flatbed-type autostereoscopic display system and its image format for encoding," *Electronic Imaging 2006*, 6055A-27, Jan. 2006.

SP