Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2009, Article ID 351452, 15 pages
doi:10.1155/2009/351452

Research Article

A Flexible Client-Driven 3DTV System for Real-Time Acquisition, Transmission, and Display of Dynamic Scenes

Xun Cao,[1, 2] Yebin Liu,[1, 2] and Qionghai Dai[1, 2]

[1] *Broadband Networks & Digital Media Lab, Tsinghua University, Beijing 100084, China*
[2] *Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing 100084, China*

Correspondence should be addressed to Xun Cao, xuncao@gmail.com

Received 1 October 2007; Revised 4 February 2008; Accepted 19 May 2008

Recommended by G. Triantafyllidis

3D experience and free-viewpoint navigation are expected to be two essential features of next generation television. In this paper, we present a flexible 3DTV system in which multiview video streams are captured, compressed, transmitted, and finally converted to high-quality 3D video in real time. Our system consists of an 8 × 8 camera array, 16 producer PCs, a streaming server, multiple clients, and several autostereoscopic displays. The whole system is implemented over IP network to provide multiple users with interactive 2D/3D switching, viewpoint control, and synthesis for dynamic scenes. In our approach, multiple video streams are first captured by a synchronized camera array. Then, we adopt a lengthened-B-field and region of interest- (ROI-) based coding scheme to guarantee a seamless view switching for each user as well as saving per-user transmission bandwidth. Finally, a convenient rendering algorithm is used to synthesize a visually pleasing result by introducing a new metric called Clarity Degree (CD). Experiments on both synthetic and real-world data have verified the feasibility, flexibility, and good performance of our system.

Copyright © 2009 Xun Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Television has greatly changed our life since its invention as early as 1920s [1]. After the era of analog TV, digital TV has become more and more popular as a revolution because of its high-quality viewing experience. However, even with high-resolution digital TV services, the observers can only watch 2D video passively. Consequently, we believe the next generation TV must have two properties: 3D effect and the ability to control the viewpoint interactively (free-viewpoint). Although stereoscopic 3D viewing techniques are almost as old as their 2D counterparts, until recent years all the conditions have enabled researchers to implement a real-time practical 3DTV system, which includes capturing and representation of dynamic scenes, compression and transmission of the data, and rendering and display on 3D devices.

Researches towards a 3DTV system started with the development of binocular stereo cameras and stereo TV just after the Tokyo Olympic Games [2]. NHK-STRL reported a stereoscopic 3D-HDTV system in 1999 [3], "sensation of reality" is mentioned by increasing spatial resolution and widening the viewing angle. Another 3D-HDTV experiment was the broadcast of the 2002 FIFA World Cup in Korea [4]. Many stereoscopic cameras were tested in this experiment. These 3DTV attempts are very valuable for evaluating visual and psychological effects but their viewpoint cannot be interactively controlled and the coding and streaming mechanism only takes into account two video streams (left and right eyes). Mitsubishi Electric Research Laboratories (MERL) setup a prototype 3DTV system with 16 cameras and a multiprojector display which can show high-resolution stereoscopic color images for multiple viewpoints without special glasses [5]. The European ATTEST project [6] demonstrates a full 3DTV processing chain in which a depth image-based rendering (DIBR) technique is interpreted. The N-view-plus-depth concept is very suitable for rendering new views and generating a 3D scene. Moreover, this scheme is backward-compatible with current 2D digital TV.

In this paper, we present a client-driven 3DTV system with a camera array over IP network. The system is scalable in terms of the number of cameras. Client-driven means the entire system pipeline is conducted according to the requirements of clients, including the cameras used to capture, the transmitting data, and the rendering views. After software-synchronized capture from the camera array, we jointly consider the coding and the rendering procedure. We employ an ROI-based coding scheme by the observation that not all part of the captured images are used to render the final result. This ROI-based mechanism can be further combined with our proposed lengthened-B-field coding scheme which can fulfill a seamless view switching. On the client side, a new virtual view is rendered depending on the client's choice. The novel rendering algorithm derives from dynamic light-field rendering (DLFR) [7] while antialiasing result is achieved by using a plane-sweeping strategy. A new metric, called Clarity Degree, is in design to implement the measurement which is very similar to those stereo algorithms. The architecture of our system is designed to meet the requirements for multiple users and is flexible enough to enable further research on 3DTV.

The main contributions of our proposed system are listed as follows.

Seamless view switching. A lengthened-B-field coding mechanism guarantees the view switching seamlessly for all users.

Scalability. The system is completely scalable in the number of cameras and displayed views.

Efficient streaming. We save the bandwidth by using an ROI-based approach which jointly considers the streaming and rendering process.

All-focus multiview video rendering. A high-quality all-focus (all-clear) rendering is achieved by introducing a new metric called Clarity Degree to distinguish clear and sharp parts of an image from those blur or ghost ones.

Flexible architecture. Our system is compatible for both 2D and 3DTV, suitable for both free viewpoint in a certain angle and 3D application, and adaptive for various users.

2. Background and Previous Work

2.1. Image and Video-Based Rendering. 3D effect and free-viewpoint navigation ability are the two key features of a 3DTV system. To provide these functionalities, traditional methods endeavor to compute three-dimensional (3D) models and do texture mapping. A drawback to this approach is the requirement for prior creation of these 3D models of objects and scenes. Such modeling is very difficult when handling dynamic scenes of complex geometry. Image-based rendering (IBR) has drawn much attention because it offers a novel alternative to conventional model-based rendering by creating a photographic realistic dynamic scene just from captured images.

Early IBR methods are derived from the plenoptic function [8]. This seven-dimensional function expresses the intensity of every light ray at every position in space (3D), in every direction (2D), at any time (1D), for every wavelength (1D). However, this description is usually simplified by omitting dimensions in practice, for instance, the wavelength, time, or spatial dimensions. A typical example of such representations is light field and lumigraph. Light field [9] makes a simplification to plenoptic function by using just two planes (4D) to represent a light ray. Other representations like ray spacing hold the similar idea as light field.

These basic concepts of IBR are easily extended to video-based rendering (VBR) by using video data as the input. Given a handful of video recordings, VBR provides us a solution for realistically rendering complex, time-varying scenes. Except for those direct transmitting and displays (using well-rectified stereo cameras), most 3DTV systems are built on the foundation of VBR techniques. Virtualized Reality [10] is probably one of earliest attempts for capturing and rendering dynamic scenes. This system is configured with 51 cameras around a 5-meter geodesic dome. A global surface representation is extracted at each frame by using the voxel coloring method. Most VBR methods can be divided into two categories: small-baseline VBR and wide-baseline VBR [11]. Since cameras are arranged too sparsely in a wide-baseline VBR technique, we take an overview of small-baseline VBR methods which are more suitable for a 3DTV system. Wilburn et al. [12] implemented an MPEG2-based light field camera array with 128 cameras to capture and store the dynamic light-field data. With this camera array, high performance imaging is achieved such as high resolution, high dynamic range, and high speed video. Synthetic aperture photography is also performed by this camera array. Yang et al. [13] developed an 8×8 grid of 320×240 camera array; they transmit only the rays necessary to compose the desired virtual view which is very similar to our ROI-based coding mechanism. A simple compression and rendering system, with its cameras located along a line, is also presented in [14] to render 3D scene interactively. A high-quality depth-based view interpolation shows us a promising result after capturing 8-view video streams with a resolution of 1024×768. Although the depth maps are generated offline, this high-quality rendering convinces us that the key techniques of interactive free-viewpoint video can be mastered in the near future. MERL proposed a real-time end-to-end 3DTV prototype system for autostereoscopic display [5], including 16 cameras are and multiprojector displays. Multiple video streams are individually encoded and transmitted over broadband network, design tradeoffs are also discussed. For multiview streaming service, a real-time multiview system with high interactivity is presented in [15], containing a 32-camera array located along an arc. With this system, users can interactively select their desired viewing directions and enjoy many exciting visual experiences, such as view switching, frozen moment, and view sweeping, in real time and with great freedom.

As a matter of fact, traditional image-based methods usually need a very dense spacing of input cameras in order to function properly. In this situation, the range of virtual viewpoints is often constrained to be close to the input camera views. Although model-based methods encounter difficulties in handling complex scenes, they have drawn much attention in recent years as hardware development such as GPU technology. Visual hull reconstructs geometry models of a scene from multiview silhouette images or video streams. Examples are image-based visual hull [16] and polyhedral visual hull methods [17]. The combination of stereo reconstruction with visual hull leads to a better reconstruction of surface concavities [18]. 3D video billboard [19] is another example of this type of method which generates time-coherent models. Stereo methods have also been applied to reconstruct and render dynamic scenes [10, 14]. Alternatively, a complete parameterized geometry model such as a skeleton model can be used to pursue a model-based approach toward free-viewpoint video [20]. These model-based approaches enable full fly-arounds in scenes without densely input cameras.

2.2. Plenoptic Sampling Theory and All-Focus Image-Based Rendering. Plenoptic Sampling theory [21] provides us a quantitative analysis of the relationships among three key elements in image-based rendering: depth and texture information, number of input images, and rendering resolution. An important conclusion from this theory is that accurate geometric scene information can greatly improve the quality of rendering result with fixed capture images. On the other hand, if the input image number increases, for example, cameras are configured more densely, the rendering resolution is also enhanced.

Shum et al. have classified IBR techniques according to how much geometric information they used [22]. In these IBR techniques, light-field rendering is a typical method because it does not require any geometric information. Isaksen et al. [7] extended light-field rendering by introducing a movable virtual focal plane (VFP) called dynamic light field rendering (DLFR). With plenoptic sampling theory, the scene objects located on the VFP of rendering will be clearly synthesized, which can be considered in focus [23]. Conversely, if the real depth of objects does not match the VFP, those objects will be rendered with unpleasing visual artifacts such as blur and ghost, this phenomenon is mentioned as out of focus. Consequently, many efforts have been made to add accurate geometric information or scene depth maps to the rendering process so as to improve the image quality. If every part of the synthesized image is clear, without the artifacts mentioned above, we call it an "all-focus" rendering. However, computing accurate geometric information is very difficult and time consuming in practice.

As discussed above, aliasing rendering occurs when there are not enough input images or lack of scene depth information. To achieve an antialiasing rendering effect, most previous efforts [7, 24] endeavor to recover an accurate model of the scene or object by offline computation and add this model information when synthesizing input videos or pictures. Some researchers have demonstrated their rendering schemes on-the-fly [23, 25, 26] by simplifying the geometrical modeling to reduce the computational time. Reference [23] illustrates an ingenious measurement to obtain an all-focus rendering called focal measurement. Some other hardware-aided method can give good results [26], effectively combing a plane-sweeping algorithm with view synthesis for real-time 3D scene acquisition. Reference [27] explicitly reconstructs photo hulls by adopting a view-dependent plane-sweeping strategy. Graphics hardware is exploited to verify the photo-consistency of each rasterized fragment. In this paper, we propose a novel metric, Clarity Degree (CD), to select clear and sharp regions from prerendered images at different depth layers. The optimal prerendering depth layers are easily obtained by following the Plenoptic Sampling theory.

2.3. Multiview Video Compression and Transmission. Efficient multiview video compression is a key component of a 3DTV system because of the vast amount of data. More and more attention has been given to research on multiview video coding. An ad hoc Group on 3D audio and video (3DAV) was founded by the MPEG community, whose main concern lies in the coding techniques of multiview video signals as well as other data such as depth, disparity, 3D geometry, or camera calibration information [28]. An overview of some early offline compression methods can be found in [2]. In multiview video coding, the traditional motion compensation in the time domain is called temporal coding while view predication between cameras is called spatial coding. Reference [14] has shown us that a combination of both temporal and spatial coding can lead to good results. A notable work proposed by [13] provides us a real-time compression for an 8 × 8 light field camera. In this work, a real-time compression and display is achieved by only transmitting rays which are needed for view interpolation. European ATTEST project promotes a compression scheme by reducing the data to a single view with a per-pixel depth map. This video-plus-depth structure is very flexible and scalable for diverse 3DTV services. It is proved that this form of data can be compressed in real time and broadcast as an MPEG-2 enhancement layer, which makes this method backwards compatible with existing broadcasting services. However, this single view plus depth mode has several disadvantages. First, difficult vision problems like visibility and occlusion cannot be easily handled with only one view of the picture; second, view-dependent appearance effects, such as reflections, cannot be revealed. As a result, this mode is further extended to N-view-plus-depth structure.

2.4. 3D Display Technologies. Many different methods of 3D displays have been presented over the last few decades [29]. Although there are various kinds of displays and project systems for generating a stereoscopic effect, most of them can be divided into the following categories: (1) holographic display, (2) volumetric display, (3) autostereoscopic display, (4) head-mounted display, (5) stereoscopic display [30]. In these displays, we believe autostereoscopic display is more suitable for individual users at present. Here, autostereoscopic means those 3D displays which can create stereoscopic effects without the help of special glasses. The type of 3D display determines the number of views which need to be rendered. If the client uses a binocular autostereoscopic display, just two views are required for left and right eyes, and if the client applies a multiview autostereoscopic display, more views are rendered. The display is the last but definitely not the least stage in a 3DTV system, it plays a significant role in providing

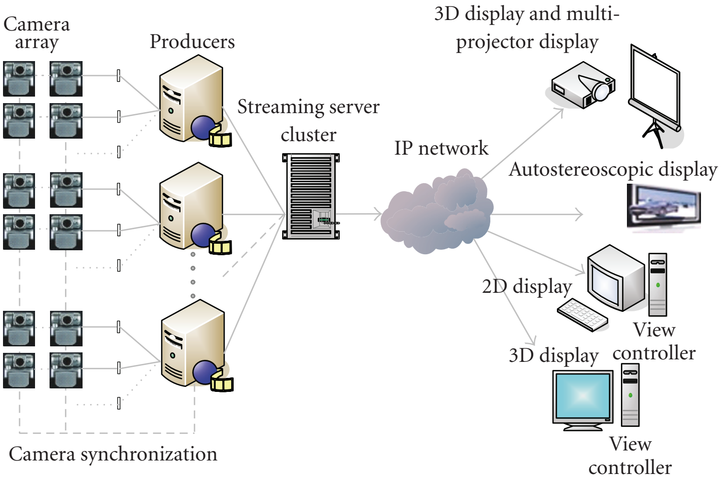


Figure 1: A flexible client-drive 3DTV system.

users with a 3D experience, and the rendering scheme is also determined by the properties of 3D displays. We believe with the development of materials, optics, psychology, and related disciplines; matured commercial 3D displays can be widely used. In our system, we use a SeeReal [31] binocular autostereoscopic display and a Bolod [32] 8-view autostereoscopic display for 3D viewing.

3. System Architecture

Figure 1 illustrates the schematic representation of our 3DTV system which consists of three main functional stages: capture part (multiview video acquisition), server part (compression and transmission), and client part (view rendering).

The capture part includes synchronization of multiview video streams, geometry and color calibration, which are accomplished on producer PCs. Data compression and streaming belong to the server part, and the client part is responsible for data decoding and view rendering. We assign the rendering work to client part with the following considerations: (1) the burden of the server will increase linearly with the number of the users if the server is involved in the rendering process; (2) our rendering algorithm is completely feasible on most commodity PCs today.

Our system is adaptive for both 2D and 3D displays since all clients share the same streaming protocol. If the stereo cameras are well rectified, we just simply decode the corresponding binocular video streams to 3D display and we can adjust our viewpoint between cameras with almost zero delay. Meanwhile, the system allows users to adjust the reproduction of binocular videos due to the fact that there is an individual preference on depth perception. Besides the stereoscopic feelings, users can enjoy other three visual experiences.

(1) Time frozen movement: users can choose to have a pause and roam in the scene for the interested people or object smoothly.

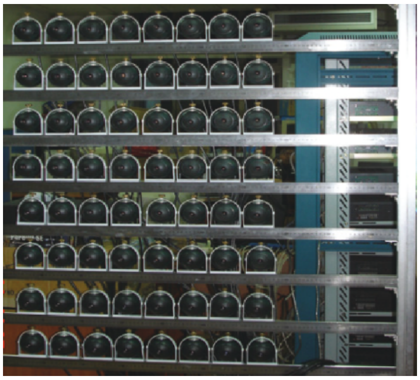


Figure 2: Our camera array.

(2) Time continual movement: users are able to change the viewing position and viewing direction as the video continues along with time.

(3) View zooming: our rendering algorithm can also provide zooming capability for users. If the user trajectories are near the camera plane, a relatively small number of frames are required to generate new views, otherwise, as for the situation of zooming in and out from the camera plane, more frames are needed.

The details of multiview video acquisition, compression, streaming, and client display are interpreted in the following subsections.

3.1. Acquisition. We setup our cameras on a regularly 2D planar array which is very similar to light field camera, but not that densely configured, providing the users with more freedom and a wider range of view. 64 BOSER BS-103F color cameras are used in our system, with a maximum 640 × 480 CCD sensor. The maximum frame rate is about 30 fps. Every 4 cameras are connected by IEEE-1394 serial bus to one of the 16 producer PCs with the same hardware configuration: Pentium-IV D 2.8 GHz and 1 GB RAM.

Since the configured BOSER BS-103F cameras do not support a trigger signal input, software is developed to synchronize the internal clocks of the producer PCs when the system starts up. This procedure is finished in 10 milliseconds, so the time variance between their clocks is no more than 5∼10 milliseconds. The total time spent by a producer PC in capturing a frame from each of its four connected cameras is no more than 5 milliseconds. The maximum variance of time between any two cameras for the light field is 15 milliseconds, which means that their frames are aligned temporally. The optical axis of each camera is roughly perpendicular to a common camera plane. The horizontal interval between cameras is about 8 cm, and the vertical distance is about 14 cm. A 14 × 14 checker board is used to calibrate the parameters of the cameras. First, the intrinsic parameters of the cameras are calibrated separately using Zhang's theory [33]. Then, every 3 × 8 subcamera array is calibrated to obtain the extrinsic parameters. When the system starts up, the producers PCs gather the raw streams captured by the cameras, and rectify the radial and tangential distortion by the intrinsic parameters. Then, aiming at accurate view rendering, both the intrinsic and extrinsic parameters are transmitted to the clients. Our module also provides an offline and an online calibration steps for color calibration.

The color calibration is very important to our system for three reasons: (1) eliminating the flicker of color in the result of the rendering; (2) improving the compression efficiency among views; (3) stereo sensation can be alleviated due to the photometrical asymmetries such as contrast or color. Since the color of the frames will change unpredictably with the environmental factors such as illumination, temperature, and the distance to the cameras, the white balance of all the cameras must be turned off, and online calibration is needed for the views involved in the interview prediction. Online calibration between cameras connected to different producer PCs is too difficult to implement because the PCI bus will be the bottleneck. Therefore our online calibration is confined to raw video sequences from the four cameras on the same producer PC. Since there is a great region overlapped among the 4 streams of frames gathered by the same PC, we simply modify the brightness of these frames so that the average brightness of their overlapped regions is equal. After the online calibration is done, an offline calibration will be carried out before initializing our system. Figure 3 shows half of a coinstantaneous group of frames captured and later rectified by our system (the sequence's name is "*Room*"). In terms of the scene, there are 4 objects located at different depth layers from the near to the distant, namely: a teddy bear, two men, a blackboard, and the wall. Such a scene can help us to understand the focus problem during the interactive changing of the virtual focal planes (VFPs) based on the DLFR.

3.2. Coding and Transmission. Three major aspects are taken into account in the coding and streaming stage: (1) seamless switching between the views, (2) coding efficiency, (3) streaming bandwidth. We make efforts to improve the above three objectives as follows. First, a "*simul-switching*" scheme is adopted to guarantee seamless view switching between multiple users simultaneously; second, a novel coding structure called lengthened-B-field coding is designed to increase the coding efficiency; finally, by jointly considering the streaming and rendering processes, independent region block coding and region-of-interest (ROI) streaming strategy are employed to reduce the streaming bandwidth.

3.2.1. "Simul-Switching" and Lengthened-B-Field Coding. Our streaming data is a 5-dimensional signal (2 dimensions for each image, 2 dimensions for the camera's position in the array, and 1 for the time axis), which presents great challenge to the storage, besides the requirement to IP network transmission. As an effective algorithm for video coding, interframe prediction which exploits the temporal redundancy is widely used in the conventional approaches. However, as mentioned above, interactive rendering requires a seamless switching among the views for multiple users simultaneously, which is described as "*simul-switching*" [34]. The main disadvantage of the conventional schemes for temporal coding is that there would be additional delay to the data transmission when switching to a particular view, because the decoder has to wait for the next I frame to decode the P or B frame. Figure 4 illustrates such a situation. Before instant *t3*, the user demands *V1* and *V2* from the streaming server, while at *t3*, the server receives the request for the views *V3*, *V4,* and *V5*. Since the frames at *t3* are all P frames, when the sequence of *V5* is added to the streaming, there is either a time delay for waiting for the next I frame or an error decoding due to the lack of reference frame for *V5*. Such a dilemma drastically affects the rendering operation.

Besides the seamless switching, another target of our coding mechanism lies in the reduction of interview redundancy. Many previous researches, for example, static light field compression, focus on exploiting the correlation of two spatial dimensions. If cameras are located densely enough, and the views are well rectified for both the geometry and color parameters, these methods can yield satisfactory compression efficiency. However, in practice, the effect of interview prediction is often below our expectation because ideal camera parameters and transmission environment cannot be obtained as well as the smaller interview correlation compared with temporal situation. These two factors make the interview predictions a subordinate part of most prediction structures. The MPEG 3DAV group is currently investigating the compression approaches with balanced temporal and spatial prediction [28]. However, these approaches are often too complicated to be adopted for practical transmission purpose, and they also bring in much inconvenience for view switching. Considering both seamless switching ability and reduction of interview redundancy, we propose a novel coding structure called lengthened-B-field to handle the difficulties mentioned above (see Figure 5).

First of all, some new terms are defined. In the multiview video data, the concept of "field" often refers to the set of

Figure 3: A snapshot of a DLF sequence (*room*) taken with our DLF rendering and streaming system.

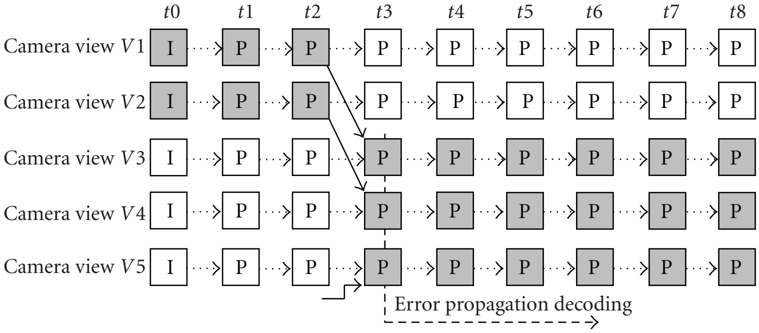


Figure 4: "Simul-switching" in dynamic light field streaming.

images captured at the same instant. The role of a "field" in the multiview data is similar to a frame in video. Based on the prediction structure between frames, new kinds of fields can be defined. Here, intracoding field is defined as the I field, in which the frames are coded without reference to any frames in other fields. The P field is known as the "refresh field", in which the frames refer to the frames in the former I field. The B field is the "bipredictive field," in which the frames may refer to the temporally former and latter I fields or P fields. When the number of B fields is predominant in the structure, for example, 30 B frames between every two P fields, the continuity of traditional temporal prediction will be broken. The compression efficiency of such prediction is still high since the camera array and the background of the scene are usually static. Moreover, in I fields, compression can be realized through coherence exploitation between the views. Figure 5(a) shows the interview prediction chains in I fields where three of the four images connected to the same producer PC are predicted from the left top image. Interfield correlation between I fields is exploited as shown in Figure 5(b).

We implement this coding scheme by a modified version of the MPEG4 XVID codec. Note that such a coding scheme does not contain multihypothesis prediction (except for the predictions of B frames), and there is no mutual communication between producer PCs. These features make this scheme suitable for real-time compression. And the multiview data can be stored in the producer PCs at this step.

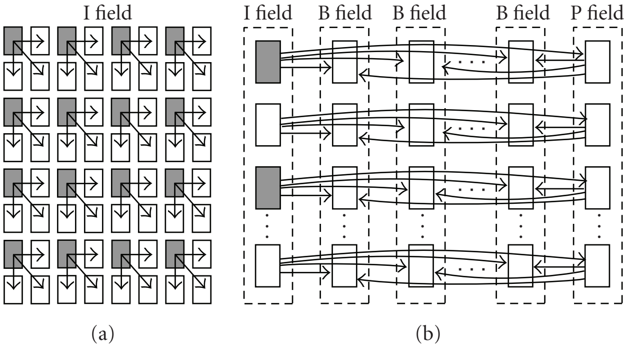


Figure 5: Prediction structure for the multiview data compression: (a) the spatial prediction structure inner I field, and (b) temporal prediction structure.

In data transmission, I fields and P fields are imperative for the clients, while the frames in B fields are optional. Thus, our coding scheme can guarantee the multiview "*simul-switching*" requirement.

3.2.2. Region of Interest Streaming. In order to further reduce the streaming bandwidth, an ROI-based streaming strategy is adopted. This strategy is feasible because most of the current rendering schemes [7, 23, 35] (with or without geometry involve, regular cameras or unstructured cameras) have the region selection procedure (from the available camera views) and the region blending process. Here, we implement dynamically DLFR as an example to illustrate how the region selection procedure can be integrated into the ROI-based streaming.

First, the data camera's aperture filter (a mask used to do texture mapping) is projected on to the virtual camera's image plane producing the region which uses samples from the data camera. Then, the data camera's aperture filter is projected on to the focal plane generating the viewing content from this data camera. Such viewing content on the focal plane is then reprojected on the data camera plane from the data camera's point of view. This reprojection produces the ROI on the data camera's image. At last, the ROI is texture mapped on to the desired image plane's region which is computed in the first step. Multiple texture mappings using both input camera data and corresponding aperture filter produce the final desired image (see Figure 6).

A more straightforward illustration of the region-of-interest for rendering procedure is shown in Figure 7.

Under the assumption that the capture range of each data camera is broad enough and the focal plane is parallel to the camera plane, the following equations can be established from this figure:

$$\frac{|\mathrm{MN}|}{|\mathrm{PQ}|} = \frac{|\mathrm{OC}|}{|\mathrm{OF}|}, \qquad \frac{|\mathrm{PQ}|}{|\mathrm{AB}|} = \frac{|\mathrm{VF}|}{|\mathrm{VO}|}. \tag{1}$$

Therefore, the region of interest |MN| can be obtained as

$$|\mathrm{MN}| = \frac{|\mathrm{AB}| \times |\mathrm{OC}| \times |\mathrm{VF}|}{|\mathrm{OF}| \times |\mathrm{VO}|}. \tag{2}$$

Based on the figure and equation above, the following conclusions concerning the region-of-interest for rendering can be made.

Conclusion 1. The region of interest for the data camera is irrelevant to the desired view direction but relevant to the view position. The closer the view position to the camera plane is, the broader the ROI will be.

Conclusion 2. The longer the distance between the camera planes to the focal plane is, the more narrow the ROI for a particular camera will be, while the number of cameras which contribute interested regions to the rendering processing will be increased.

3.2.3. Independent Region Block Coding. Besides the ROI-based streaming, we implement an independent region block coding. Although this coding scheme lowers the coding efficiency since block correlation is not explored, the entire streaming bandwidth is reduced by this method because not all blocks are transmitted.

An image can be completely and regularly partitioned by several region blocks which are composed of some macroblocks. As for a P-field image at 320 × 240 resolution, there are 24 partition modes and the corresponding region blocks can be of size $16m \times 16n$ ($m = 1, 2, 4, 5, 10, 20$, $n = 1, 3, 5, 15$). Figure 8 illustrates the partition of region block at size 80 × 80. The required area for streaming under a particular mode is the minimal set of region blocks that can cover the region of interest.

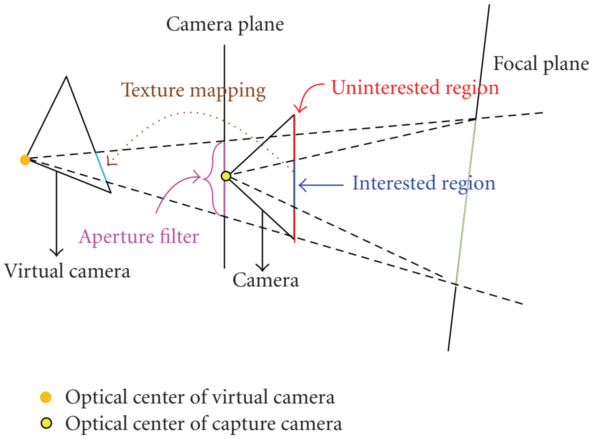


Figure 6: Region selection in rendering procedure.

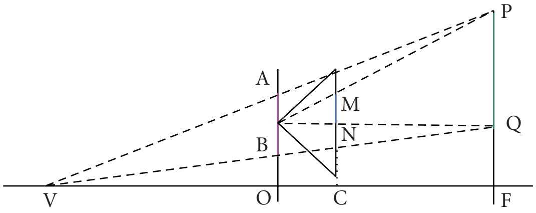


Figure 7: Geometry in region selection of rendering.

Each region block is independently coded similar to "slice partition" mechanism in H.264/AVC. Unlike "slice partition," our block partition achieves random access operation to any region block when it is combined with the lengthened-B-field coding mechanism. Through the decorrelation of motion vectors on the edge of region blocks and the insertion of synchronization bits before each of them, all data in P fields are coded as independent region block streams to economize the streaming bandwidth. Once the partition mode and the camera parameters are determined, the required region block streams for the particular virtual view rendering can be computed and recorded as lookup tables using the model in Section 3.2.2. It must be noted that fine partition may result in economized area for transmission but worse compression ratio. Hence, the choice of region block size must be based on rendering algorithm and coding characteristic.

3.2.4. Streaming Strategy. Figure 9 shows the streaming strategy of our system. Once there are clients requesting a streaming service, all the producer PCs send their 4 compressed streams to the streaming server, which stores the content for the last 2 seconds in the buffer. When users change to different viewpoint or view direction, the client part of our system calculates and sends the users' requests through the feedback channel to the streaming server. If these requests are received by the streaming server, the server will switch to the B field stream, and then send them to the clients through the data channel. Here, frames in I and P fields are the imperative, and it is only after the transmission of them will the ones in B fields be transmitted. Since the buffer delay is uncorrelated to user control delay, only the network's round trip time will be experienced by the users, when they change the viewpoint or direction. Moreover, this streaming system is also compatible with other kinds of display, including the 2D display, stereoscopic display, and autostereoscopic display. And it only needs to send the requested streams for any particular client, without any consideration on the display or the rendering algorithm.

3.3. Real-Time Rendering

3.3.1. All-Focus Rendering. The rendering scheme of our system derives from the plane-sweeping method by introducing a new measure metric called clarity degree. We first render several images while changing the virtual focal plane (VFP) at a given viewpoint, and then detect the clear parts of those images and integrate them into one final result which would be all clear without unpleasant visual artifacts. These VFPs can be preset with the knowledge or assumption of the maximum and minimum scene depth, rather than complicated computer vision algorithms. Plenoptic Sampling theory [21] tells us that we can render more satisfactory images with more knowledge about the scene geometry, for example, depth map. The optimal render depth can be computed as follows:

$$\frac{1}{Z_{\text{opt}}} = \frac{1/Z_{\min} + 1/Z_{\max}}{2}, \tag{3}$$

where Z_{opt} is the optimal rendering depth, $Z_{\min}$ and $Z_{\max}$ denote the minimum and maximum depth of the scene, respectively. As a result, if we have prerendered the original scene at adequate depth layers, a good rendering result can be achieved. However, more prerender depth layers also cause a linear increase of time consumption. Thus, there is a tradeoff between the visual effect and time consumption in the rendering process.

In this strategy, the key problem is to formulate a metric to distinguish the clear and sharp parts in an image from the blurred and ghosted regions. In our scheme, we use the metric proposed in [35] called clarity degree. Clarity Degree (CD) tells us the extent of clarity of a certain image region by evaluating the sharpness in the energy domain. Clarity Degree (CD) is quantitatively defined in local regions by the index called mean change energy (MCE) criterion, which is very easy to calculate to tell an aliasing region from a clear one. In a certain image region, we can measure the Clarity Degree by following a simple principle: the higher the MCE value is, the clearer this image region will be. Determined by most IBR algorithm's properties, major factors affecting clarity and sharpness of single-depth rendered images are the existence of blur and ghost (double image); see Figure 10. Thus, it is required that the proposed metric has a strong ability to distinguish the clear part of an image from the blurred and ghosted regions, especially change-intensive regions which are much more noticeable for human vision system (HVS).

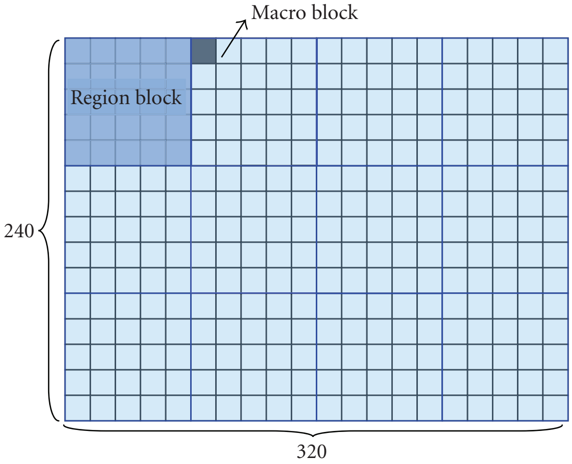


FIGURE 8: 80 × 80 region block partition for ROI-based streaming.

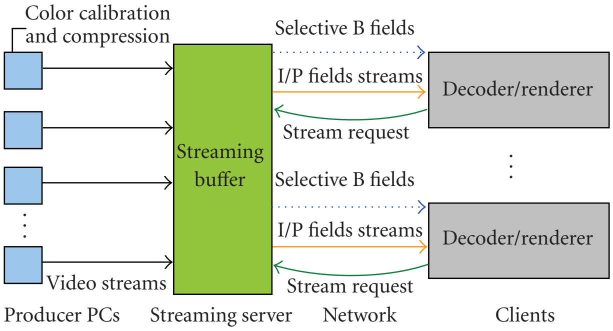


FIGURE 9: Block diagram of our streaming strategy.

We assume a set of pixels are belonging to a given image region $(x, y) \in R$, then the pixel's differential magnitude is used to reflect the changes happening within it, namely $|\nabla R(x, y)|$. To reduce interference of noise, a threshold T_{ch} is set to remove indistinctive changes. With this threshold, all pixels in $R(x, y)$ can be classified into two categories: if $|\nabla R(x, y)| > T_{\text{ch}}$, pixel (x, y) is called a "change point" where prominent changes happen; otherwise, if $|\nabla R(x, y)| \leq T_{\text{ch}}$, (x, y) is called a "smooth point." All change points compose a change set: ChSet.

By summing the differential magnitude at all change points in R, we obtain the total change energy (TCE) which reflects the global extent of major changes in this region. Generally, a sharp rendered result is inferred by a large TCE value. Unfortunately, ghosted effects, different from blur situation, produce double-edge phenomena which increase the TCE value, making it a difficulty to tell blur and ghost artifacts at the same time.

We solve this dilemma by averaging the TCE onto all change points. Mean change energy (MCE) is introduced as follows:

$$\text{MCE} = \frac{\sum_{(x,y)\in R} |\nabla R(x, y)|}{\|\text{ChSet}\|}, \tag{4}$$

where $\|\text{ChSet}\|$ indicates the number of elements in ChSet.

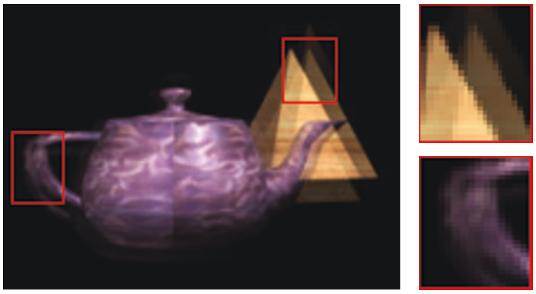


Figure 10: Artifacts in a single-depth rendered image. Right-top: the ghost or double image, right-down: blur situation.

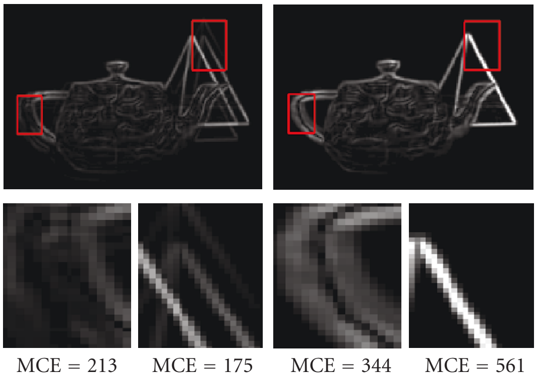


Figure 11: Clarity degree measurement by MCE. Top-left: the derivative of a single-depth rendered image; Top-right: the derivative of an all-clear image. The second row: local regions and their MCE values. The higher MCE reflects better clarity.

Clarity Degree is measured by calculating the MCE value. This measurement is very effective at selecting clear regions out of both blurred and ghosted ones, because blurred blocks are distinguished through cumulating prominent change energies with threshold, and ghosted blocks are told apart through energy averaging. Figure 11 gives an example.

With this robust measurement, we then use a block-based synthesis scheme which is designed to be compatible with current video coding techniques. The entire rendering procedure is performed by pure image processing methods, rather than complex depth estimation or iterative rendering. Moreover, this novel rendering algorithm can be easily integrated with video coding techniques. The decoded video streams are first analyzed by extracting a motion vector map. The scene structure is inferred through this motion vector map by the observation that the motion vectors can be treated as a depth cue. In a group of picture (GOP) of a compressed video, a lot of regions of successive frames are very consistent with the first frame (I frame).

As a result, much computation time in all-clear rendering can be saved by performing the motion extraction. The flow chart of this rendering scheme is illustrated in Figure 12.

When a multiview video stream comes, we first choose the camera with smallest distance to the virtual rendering viewpoint as a reference camera. In a GOP, the first frame (I frame) is synthesized using the exact same method in [35].

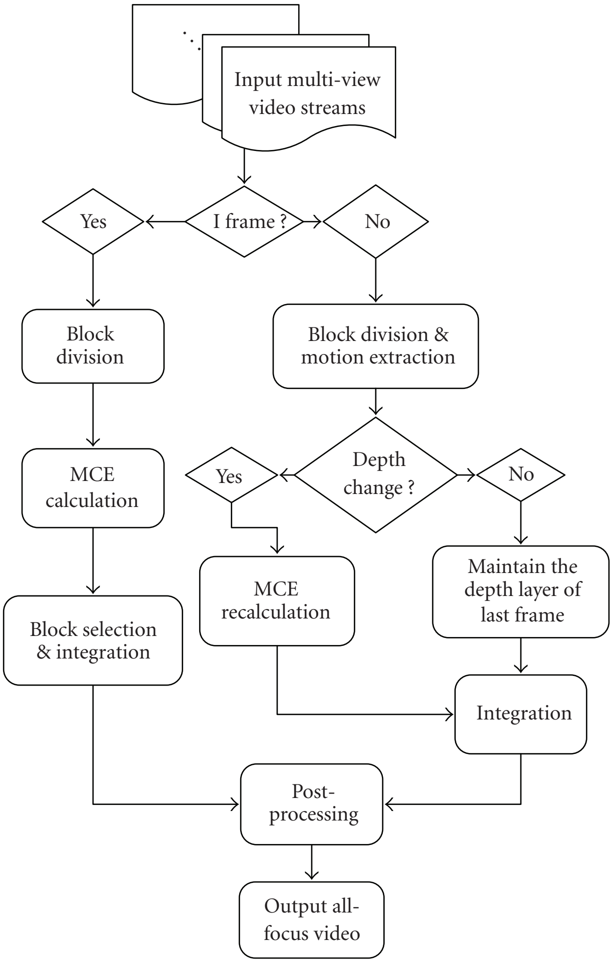

Input multi-view video streams
I frame ?
Yes
No
Block division
Block division & motion extraction
MCE calculation
Depth change ?
Yes
No
Block selection & integration
MCE recalculation
Maintain the depth layer of last frame
Integration
Post-processing
Output all-focus video

Figure 12: Flow chart of propose rendering scheme.

Step 1. Given a virtual rendering viewpoint, we first render the image at different VFP using the algorithm in [7]; the VFP is set with optimum value according to Plenoptic sampling theory.

Step 2. Divide this frame into blocks, calculate the MCE value in every block at different VFP, and choose the block with highest MCE value as the clearest one. The block size can be preset or set adaptively.

Step 3. Integrate all the clear blocks into a final image. A postprocessing is also applied to alleviate the block effect.

The frames following the I-frame are synthesized more efficiently. After the block division, we will first decide whether this block should be recomputed. Since motion vectors can be easily extracted during the decoding processing, we just simply record the motion vectors of both x-axis and y-axis. Then, we use the following index as a clue for depth change:

$$d(i,j) = \sqrt[2]{\mathrm{mv}(i,j)_x^2 + \mathrm{mv}(i,j)_y^2}, \qquad (5)$$

where $d(i, j)$ is the depth cue of the block (i, j); $\mathrm{mv}(i, j)_x$ and $\mathrm{mv}(i, j)_y$ stand for the x-axis and y-axis motion vector of block (i, j), respectively.

We assume there exists a depth change if the $d(i, j)$ is larger than a threshold thersh_d. Contrarily, if the $d(i, j)$ is smaller than thresh_d, we just maintain the VFP of last frame for this block. Hence, no more computation is needed for most blocks in the following frames, which saves a lot of time. Finally, we composite both recalculated and maintained blocks into a whole image, which will be all clearly rendered cooperating with a block effect reduction.

3.3.2. Parameters Discussion

Block size. The block size in all-clear synthesis is better set as the same as the size in the video coding, in this way, the motion vector can be extracted more conveniently. But this is not necessary in some case like homogeneous areas, in these areas, block size can be set larger while the motion vector map can be computed from several small blocks.

thresh_d. The threshold which determines whether a recalculation is needed in motion extraction can be set interactively. A larger threshold can lead to a quicker processing but a less accurate rendering result while a smaller threshold gives a better image quality but more computation.

Motion vector map. In most cases, the extracted motion vector map can be approximated as a depth cue. However, there are some special situations where the approximation does not hold. This happens when the motion is either too rapid in terms of camera rotation or in the case of camera zoom. Since in most IBR situations, cameras are configured stationary and the zoom operation is seldom adopted, the assumption satisfies most objective requirements.

4. Experimental Results and Discussions

We test our system on campus's network. 9-view video streams are used for rendering. Each client is equipped with a Pentium-IV D CPU 2.4 G, 1 GB RAM, and commodity graphic card PC which can decode and render these 9 camera streams at 30 fps in real time (without all-focus rendering, the frame rate with all-focus rendering is about 22 fps). The user can feel a time delay within only 0.1~0.9 seconds. The producer PCs are powerful enough for 4 camera capturing, color calibration, and data compression. Our streaming server has a configuration of Pentium-IV D CPU 3.0 G and 1 GB RAM, Figure 13 illustrates the emulated computational cost changing with the number of users. Also, we have examined transmission quality as the number of user increases. The output of our streaming server is Gigabits network, and when user number is lesser than 16, lost rate will be lower than 0.5%.

4.1. Coding and Streaming. The efficiency of intraprediction, conventional temporal prediction, interview prediction, and our proposed lengthened-B-field (with the 100 B fields between two P fields) method has been illustrated in Figure 14. As we see from Figure 6, even if the color calibration is satisfactory enough, the efficiency of spatial prediction will still be much lower than the temporal method. For the lengthened-B-field prediction, as the interval between the coding frame and the former I(P) field increases to 50, the prediction efficiency becomes rather low but it will turn higher as the coding frame gets nearer to the next I(P) field. It is clear that the lengthened-B-field prediction has a better performance than interview prediction.

Figure 15 shows the simulation results of the required transmission bandwidth for each user versus the average number of views for rendering (horizontal coordinate) when

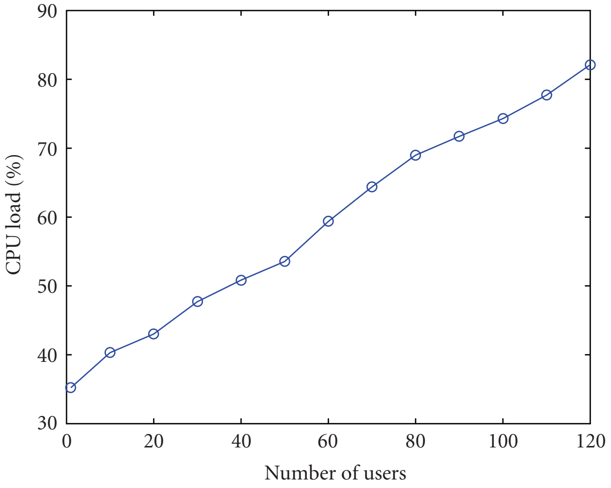


Figure 13: Computation cost versus number of users.

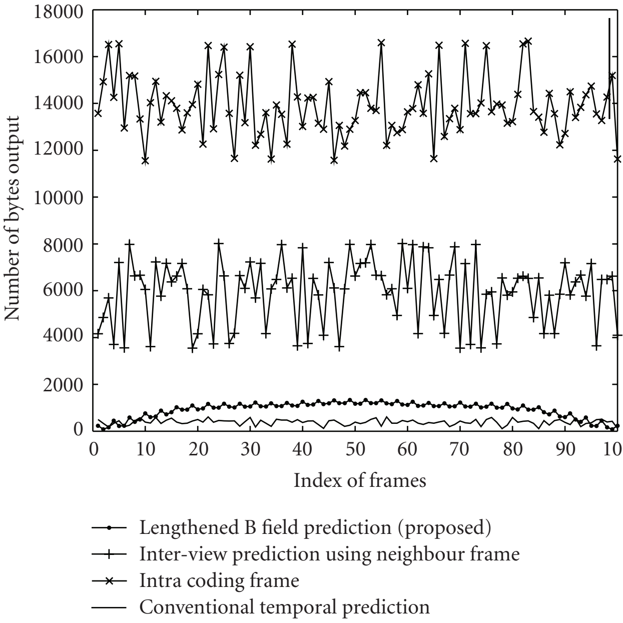


Figure 14: Comparison of the compression efficiencies for *room* sequence.

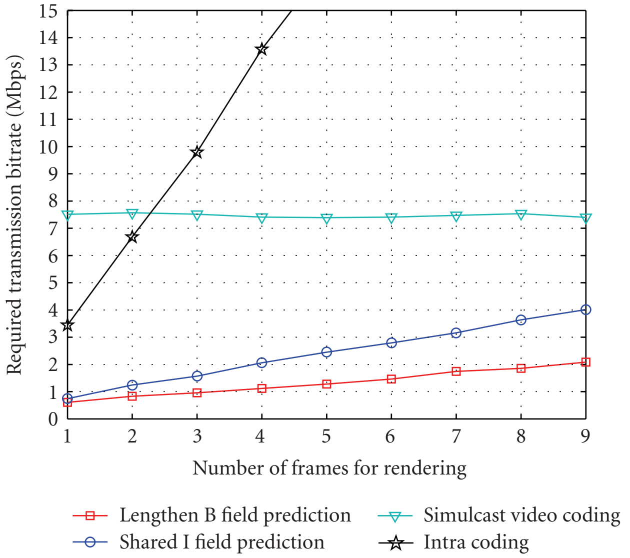


Figure 15: Average required transmission bandwidth changing with the number of views required for rendering. All these schemes deal with the room sequence, with 64 sequences from different views of camera. For the lengthened-B-field prediction, the interval between the I fields is set to 120, and the number of successive B fields between 2 P fields (or an I field and a P field) is set to 30. For the shared I field coding scheme, the interval between I fields is set to 120.

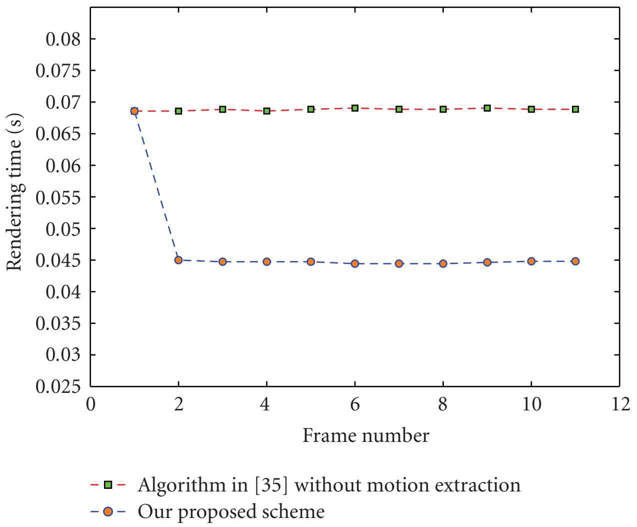


Figure 16: Rendering time comparison with [35].

Table 1: Comparison on rendering image quality.

Dataset (PSNR/DB)	Akko & Kayo	Tsukuba
Rendering at one depth	35.0784	30.7258
Rendering with depth map	35.4306	31.3825
Our method	35.2376	31.9638

the average image quality ranges from 35.8 dB to 36.1 dB. Four different coding schemes are compared including intracoding scheme, conventional simulcast video coding scheme, share-I-field coding scheme, and lengthened-B-fieldcoding scheme. For the intracoding scheme, only the frames selected by the clients are transmitted, and the bit rate presents a linear increase with the number of views required by the rendering method. In the simulcast situation, all the 64 view streams have to be transmitted, with no regard to the rendering methods. And in the shared I field coding scheme, except for the I frames, all the other frames are P frames, and are predicted only by referring to the I frames. The details are reported in our previous work [34]. From the figure we can see that our scheme outperforms other three. In this comparison, viewpoints are assumed to be set near the camera plane, thus only four views are required for the rendering. In this case, the bandwidth required is 1.2 Mbps, which is completely acceptable for the users on broadband IP network.

4.2. Rendering Results. Both synthetic and real world data are used to evaluate our rendering scheme. Two synthetic scenes (see Figures 18(a)–18(g)) are both produced by 3DMAX 7. One of the real-world data is the “Room” data set mentioned above (see Figures 18(h)–18(k)); the other real data, “Akko and Kayo,” is published by Tanimoto Laboratory, Nagoya University.

The first synthetic scene is a challenging test because the grid texture is very sensitive to artifacts, which are illustrated in all the single-depth rendered images; see Figures 18(a)–18(c). Our algorithm demonstrates a perfect performance as seen in Figure 18(d). The teapot scene verifies the ability of our Clarity Degree Measurement to distinguish both blur and ghost. Our rendering scheme is still verified very powerful to create all-focus and sharp results by two more complex real-world scenes containing more details and textures.

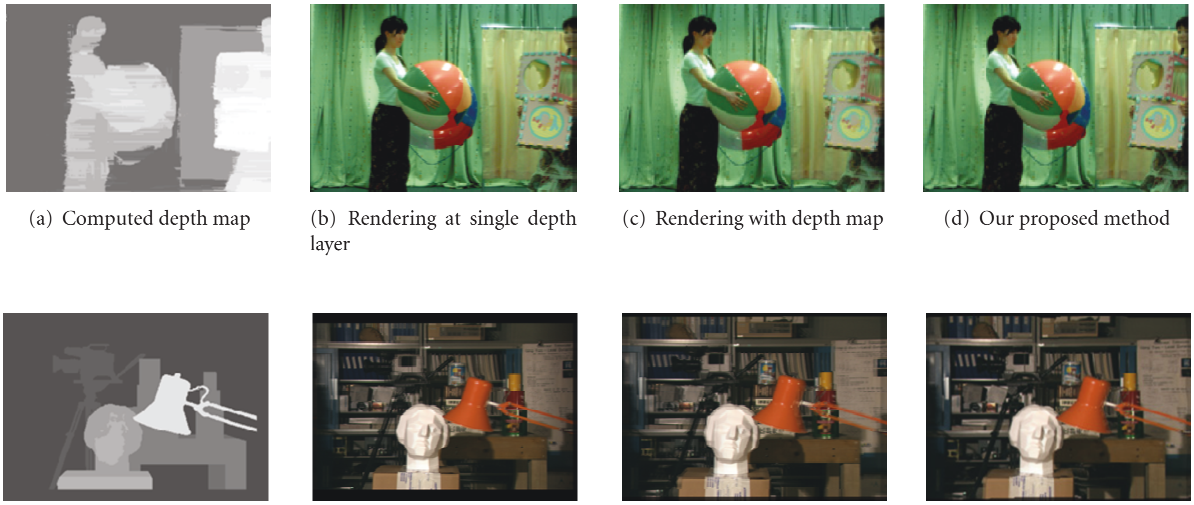
Furthermore, a time consumption comparison with the algorithm [35] is performed. Figure 16 illustrate the rendering time of the first eleven frames of a GOP, using the method in [35] and our proposed scheme. The total time consumption is saved about 32% by the proposed method.

In the “*Room*” scene, we can find that the objects located at different depth layer are all clearly rendered, but block effect as well as some ghosts still exists because only three VFPs are prerendered. The “Akko and Kayo” data gives better results because more VFPs are prerendered but is cost much more time. As a result, the tradeoff between rendering effect and time consumption must be considered.

4.3. Rendering Quality Evaluation. In this section, we compare our rendering scheme with other methods (see Figure 17 and Table 1). Both “Akko and Kayo” and “Tsukuba” datasets (published by University of Tsukuba “head and lamp” data set) are used in this experiment. In the “Akko and Kayo” sequence, we compute the pixel-level depth map using a correspondence strategy described in [36], while in the “Tsukuba” test, ground-truth depth information is added. We adopt peak signal-to-noise ratio (PSNR) as a metric to evaluate the rendering quality. We have to state that PSNR cannot reflect all the aspects in

TABLE 2: Comparison of our system with the state-of-the-art 3DTV systems.

	Systems					
Items	Interactive multiview video system [15]	Distributed light-field camera [13]	High-quality free-viewpoint video system [14]	MERL 3DTV system [5]	DIBR approach [6]	Our system
Applications	Network interactive entertainments, sports, and so forth.	Interactive video-based rendering.	High-quality archival of dynamic events and instructional videos.	Real-time 3D viewing for multiple users.	3DTV broadcasting and virtual view rendering.	Interactive video-based rendering and 3D display for large number of IP network users.
Camera configuration	32 × 1 1D arc-placed, sparsely	8 × 8 2D planar densely	8 × 1 1D linear (comparatively dense)	16 1D linear (interval not mentioned)	N 1D cameras (interval not mentioned)	8 × 8 2D planar (comparatively dense)
Rendering method	Not mentioned	DLFR [7]	Layered representation and blending [14]	Unstructured lumigraph rendering [37]	Depth image-based rendering [6]	All-focus image-based rendering
Depth map generation	Not mentioned	interactively set VFP with focus problem	Segmentation -based stereo matching	interactively set VFP with focus problem	HRM algorithm [38] or special camera	Preset depth layers selected by CD metric
Data compression	Single-view video coding and single-moment video coding	Individual video compression	Exploit both temporal and spatial redundancy	Individual video coding	MPEG2 with depth map as advanced layer	Lengthened-B-field coding
Image resolution	640 × 480	320 × 240	1024 × 768	1300 × 1030 1024 × 768	Digital TV format (720 × 576)	320 × 240 640 × 480
Frame rate	30 fps	15 fps	5 fps	Maximum 12 fps	25 fps	Maximum 22 fps
Streaming Bandwidth	About 2 Mbps	About 4 Mbps	Not mentioned	4.3 Mbps at certain condition	About 110 kbps with one view	Below 1.2 Mbps
User support ability	Serve a large number of users.	Serve limit number of people.	Not mentioned	Serve a large number of users.	Serve a large number of users.	Serve a large number of users.
Real-time property	Yes	Yes	No	Yes	Yes	Yes



(a) Computed depth map

(b) Rendering at single depth layer

(c) Rendering with depth map

(d) Our proposed method

(e) Grounded truth depth map

(f) Rendering at single depth layer

(g) Rendering with depth map

(h) Our proposed method

FIGURE 17: (a)–(d) Comparison on "Akko & Kayo" dataset with computed depth map. (e)–(h) Comparison on "Tsukuba" dataset with ground-truth depth map.

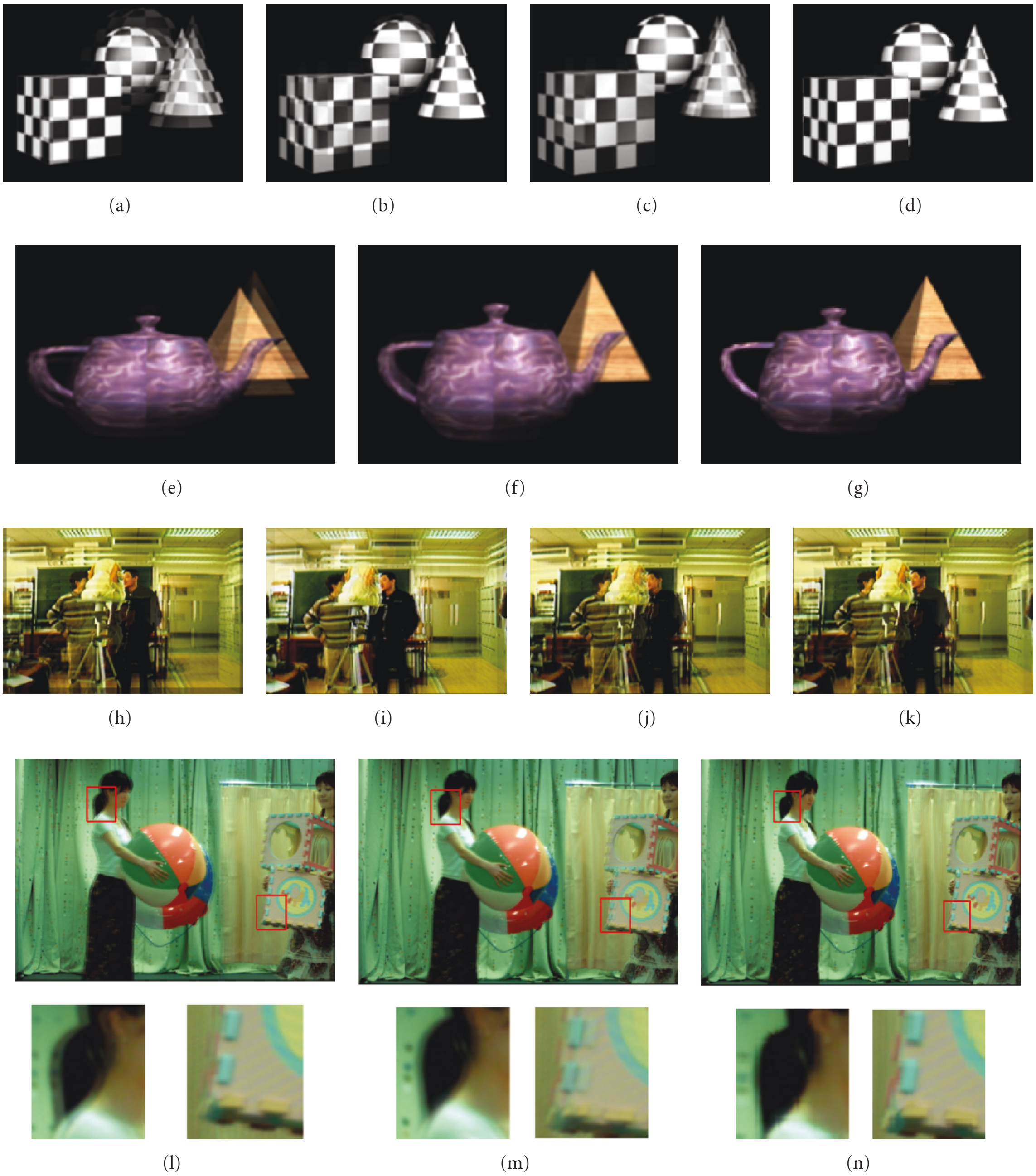


Figure 18: (a)–(d) Synthetic scene of objects with grid textures. (e)–(g) Synthetic teapot scene. (h)–(k) Real-world scene: "Room." (l)–(m) Real-world scene: "Akko and Kayo." In each of the four scenes, from left to right, there are: the result rendered with const depth at nearer objects; the result rendered with const depth at farther objects; the all-clear images are in the last column.

multiview rendering image quality. For instance, occlusion areas cannot be evaluated by PSNR. From the comparison, we can see that our method performs almost the same quality as rendering with depth maps. Moreover, our blockwise method is suitable for a simple and quick implementation.

4.4. Comparison with Previous Systems. In this section, we compare our proposed scheme with several typical previous 3DTV systems.

The interactive multiview video system [15] is a distinct one which cannot be counted as a free-viewpoint video system strictly since it does not render smooth virtual views. The system employs 32 1D arc-type cameras to capture the dynamic scenes and streams only one view at a time to the clients on IP network. The advantage of this system lies in its broad horizontal field of view and economized network transmission bandwidth. The distributed light field camera system [13] is probably the first real-time interactive dynamic light field streaming system. The view rendering procedure is assigned to the server; such configuration can be called "interactivity on the server." Network transmission bandwidth is saved for only one virtual view. However, since the server must deal with rendering and compression for each client, the system cannot serve as many users as traditional single-view video streaming systems. The high- quality video system using a layered representation [14] is a remarkable attempt for its top-quality and real-time virtual view rendering. Since the work aims at offline processing of the captured dynamic scenes, it could not be introduced into the live streaming framework currently. MERL presents an end-to-end distributed scalable 3DTV system with high- resolution (1024×768) display, which is perhaps the first real-time end-to-end 3D TV system with enough views and resolution to provide a truly immersive 3D experience. But only temporal encoding of individual video streams is adopted. DIBR [6] is a very flexible technique to be a bridge between 2D and 3D TV system. Depth computing plays a significant role in most previous researches. To validate the feasibility of 3D TV service in different situations, we develop a real-time interactive rendering and streaming system over broadband IP network for multiple users. The comparisons are detailed in Table 2.

5. Conclusions and Future Works

In this paper, we proposed a flexible client-driven 3DTV system to offer multiple users a 3D view experience over IP network. After introducing relevant concepts and technical challenges, we provide a novel streaming and rendering mechanism in which both seamless view switching and antialiasing rendering are achieved. Bandwidth is saved, and view rendering is accelerated by jointly considering the coding and rendering procedures. The proposed ROI-based transmission strategy is based on the observation that not all part of the decoded pictures is used for view synthesis. And according to the experiment of our system, which is still a prototype, the bandwidth required for streaming is below 1.2 Mbps when the viewpoint is near the camera plane. Such bandwidth is completely feasible to the users on broadband IP network. Multiple users can enjoy a seamless view-switching simultaneously in this system by adopting a lengthened-B-field coding method. Another special feature of our system is that we can render an all-focus virtual view through a few preset depth layers. The blockwise implement of the algorithm can be further combined with video coding techniques to accelerate the rendering process. We believe that with the development of camera and 3D monitor techniques, and as the computer processing power becomes stronger and networks bandwidth becomes broader, the commercial and home-based implementation for free viewpoint control and 3D display over IP network will not be too far in the future.

Acknowledgments

This work is supported by the Distinguished Young Scholars of NSFC (no. 60525111) and 863 Program (no. 2007AA01Z332) and the key project of NSFC (no. 60432030).

References

[1] http://www.tvhistory.tv/.

[2] B. Javidi and F. Okano, *Three-Dimensional Television, Video, and Display Technologies*, Springer, Berlin, Germany, 2002.

[3] NHK Annual Report, 3-D Hi-vision (HDTV) System, 1999.

[4] N. Hur, C.-H. Ahn, and C. Ahn, "Experimental service of 3DTV broadcasting relay in Korea," in *Three-Dimensional TV, Video, and Display*, vol. 4864 of *Proceedings of SPIE*, pp. 1–13, Boston, Mass, USA, July-August 2002.

[5] W. Matusik and H. Pfister, "3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 814–824, 2004.

[6] C. Fehn, P. Kauff, M. Op de Beeck, et al., "An evolutionary and optimised approach on 3D-TV," in *Proceedings of the International Broadcast Conference (IBC '02)*, pp. 357–365, Amsterdam, The Netherlands, September 2002.

[7] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 297–306, New Orleans, La, USA, July 2000.

[8] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computation Models of Visual Processing*, M. Landy and J. A. Movshon, Eds., pp. 3–20, MIT Press, Cambridge, Mass, USA, 1991.

[9] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 31–42, New Orleans, La, USA, August 1996.

[10] T. Kanade, P. Rander, and P. J. Narayanan, "Virtualized reality: constructing virtual worlds from real scenes," *IEEE Multimedia*, vol. 4, no. 1, pp. 34–47, 1997.

[11] M. Magnor, M. Pollefeys, G. Cheung, W. Matusik, and C. Theobalt, "Video-based rendering," in *Proceedings of the 32nd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '05)*, Los Angeles, Calif, USA, July-August 2005.

[12] B. Wilburn, N. Joshi, V. Vaish, et al., "High performance imaging using large camera arrays," in *Proceedings of the 32nd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '05)*, pp. 765–776, Los Angeles, Calif, USA, July-August 2005.

[13] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera," in *Proceedings of the 13th Eurographics Workshop on Rendering*, pp. 77–86, Pisa, Italy, June 2002.

[14] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proceedings of the 31st International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, pp. 600–608, Los Angeles, Calif, USA, August 2004.

[15] J.-G. Luo, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pp. 161–170, Singapore, November 2005.

[16] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-based visual hulls," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 369–374, New Orleans, La, USA, July 2000.

[17] J. S. Franco and E. Boyer, “Exact polyhedral visual hulls,” in *Proceedings of the 14th British Machine Vision Conference (BMVC ’03)*, vol. 1, pp. 329–338, Norwich, UK, September 2003.

[18] M. Li, H. Schirmacher, M. Magnor, and H.-P. Seidel, “Combining stereo and visual hull information for on-line reconstruction and rendering of dynamic scenes,” in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pp. 9–12, St. Thomas, Virgin Islands, USA, December 2002.

[19] M. Waschbüsch, S. Würmlin, and M. Gross, “3D video billboard clouds,” in *Proceedings of the 28th Annual Conference of the European Association for Computer Graphics (EUROGRAPHICS ’07)*, Prague, Czech Republic, September 2007.

[20] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, “Free-viewpoint video of human actors,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 569–577, 2003.

[21] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong, “Plenoptic sampling,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’00)*, pp. 307–318, New Orleans, La, USA, July 2000.

[22] H.-Y. Shum, S. B. Kang, and S.-C. Chan, “Survey of image-based representations and compression techniques,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, 2003.

[23] K. Takahashi, A. Kubota, and T. Naemura, “Focus measurement and all in-focus image synthesis for light-field rendering,” *Systems and Computers in Japan*, vol. 37, no. 1, pp. 1–12, 2006.

[24] H.-Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C.-K. Tang, “Pop-up light field: an interactive image-based modeling and rendering system,” *ACM Transactions on Graphics*, vol. 23, no. 2, pp. 143–162, 2004.

[25] T. Naemura, J. Tago, and H. Harashima, “Real-time video-based modeling and rendering of 3D scenes,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 66–73, 2002.

[26] R. Yang, G. Welch, and G. Bishop, “Real-time consensus-based scene reconstruction using commodity graphics hardware,” in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications (PG ’02)*, pp. 225–235, Beijing, China, October 2002.

[27] M. Li, M. Magnor, and H.-P. Seidel, “Hardware-accelerated rendering of photo hulls,” in *Proceedings of the 25th Annual Conference of the European Association for Computer Graphics (EUROGRAPHICS ’04)*, Grenoble, France, September 2004.

[28] A. Smolic and H. Kimata, “Report on 3dav exploration,” MPEG Document, ISO/IEC JTC1/SC29/WG11, N5878, July 2003.

[29] M. Starks, “3D for the 21st century: the Tsukuba expo and beyond,” 3DTV Corporation, 1996.

[30] L. Onural, A. Smolic, T. Sikora, M. R. Civanlar, J. Ostermann, and J. Watson, “An Assessment of 3DTV Technologies,” http://www.3dtv-research.org/.

[31] http://www.seereal.com/en/autostereoscopy/technology_cn.php.

[32] http://www.bolod.com/.

[33] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[34] Y. Liu, Q. Dai, and W. Xu, “A real time interactive dynamic light field transmission system,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’06)*, pp. 2173–2176, Toronto, Canada, July 2006.

[35] X. Cao, S. Xue, and Q. Dai, “All-clear image based synthesis using clarity degree,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’07)*, vol. 1, pp. 469–472, Honolulu, Hawaii, USA, April 2007.

[36] S. Birchfield and C. Tomasi, “A pixel dissimilarity measure that is insensitive to image sampling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, 1998.

[37] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, “Unstructured lumigraph rendering,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’01)*, pp. 425–432, Los Angeles, Calif, USA, August 2001.

[38] N. Atzpadin, P. Kauff, and O. Schreer, “Stereo analysis by hybrid recursive matching for real-time immersive video conferencing,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 3, pp. 321–334, 2004.