# A RECEIVER-DRIVEN MULTICASTING FRAMEWORK FOR 3DTV TRANSMISSION

*Engin Kurutepe, M. Reha Civanlar, A. Murat Tekalp*
*School of Engineering, Koç University, Istanbul, Turkey*
*{ekurutepe, rcivanlar, mtekalp}@ku.edu.tr*

## ABSTRACT

*Contemporary television and video experience is not interactive and users have little or no choice over their viewing angle in the scenes they watch. There is a demand for a real 3-D interactive experience which would allow users to view scenes through virtual cameras chosen by their head and eye locations as in real life. However, among other issues the amount of bandwidth required to transmit very large Image Based Rendering (IBR) representations of the scene to end users is still an unsolved problem.*

*In this paper we propose a novel networking scheme to enable users to automatically stream only the parts of the light field representation, which will be used to render the current viewpoint. The proposed system also incorporates prediction of future views to prefetch streams, which are likely to be needed in the near future as the viewpoint changes over time.*

## 1. INTRODUCTION

Even though a number of contemporary DVD's provide user with the ability to choose viewing angles, this only involves a selection between prerecorded static camera views and does not provide a really interactive 3-D experience. To achieve such interactive experience in 3-D the user should be able to view the reconstruction of the scene from all possible angles as if they are actually in the scene. This requires the capture of all necessary optical information from the real world scene and an efficient representation of this information for storage and transmission. There are two popular approaches for the solution of this representation problem. Texture mapping geometric models of the objects in the scene is commonly used in computer graphics applications to render views of computer-generated objects. However the computational complexity of rendering high-resolution photo-realistic novel views from a 3-D scene is highly dependent on the scene geometry and usually very high. Moreover accurately capturing 3-D geometry of real world objects is still an unsolved problem. The other approach, Image Based Rendering, aims to generate novel views of the scene from captured images. The idea behind IBR systems is the seven dimensional plenoptic function, which describes all potentially available optical information in some region [2]. Various IBR systems, such as light fields [3] or the lumigraph [4], are simplifications of this plenoptic function. Pure IBR systems do not assume an explicit 3-D model of the scene but some geometric information about the scene, usually in a depth map form, is sometimes used to reduce the sampling rate in camera plane and to improve reconstruction quality.

Rendering novel views from an IBR representation requires much less computational resources than rendering views by texture mapping geometric models. However the reconstruction quality of IBR systems depends on the sampling density in the camera plane. As a result IBR representations for a good reconstruction quality require a high number of cameras to capture the scene and generate huge amounts of data. Previous research [1, 3, 6] has shown that raw data for a single static light field can reach several hundred megabytes or even gigabytes for high-resolution examples. Light fields contain highly coherent data and high compression rates in the order of 500:1 and 1000:1 have been reported for static light fields [6,7], but these high compression rates come at the cost of the ease of interactive viewing due to the dependencies created between light field images.

The data rate problem is even more important for light field videos which consist of 30 static light fields for every second. This roughly corresponds to 30GB raw data each second if we assume each frame to be a high quality light field of about 1GB. There is little research on compression of dynamic light fields but it appears unlikely to expect compression algorithms to be developed, which would reduce the data rate to levels that are feasible to stream over current domestic broadband connections. Therefore there is a need for a novel networking scheme to be able to send the light field videos over existing broadband connections.

Multicasting is widely used for transmitting ordinary video streams over the Internet. It involves routing packets from a set of servers to a set of receivers, such that each packet is forwarded to all receivers interested in receiving that packet. This saves bandwidth over the whole network by avoiding sending the same packet more than once to several receivers separately. As McCanne et al discuss in [8] multicasting principle can be further adapted to transmission of multimedia data by multicasting the data in multiple layers and giving the control over which layers to subscribe to receivers. In video transmission these layers usually correspond to a base quality layer and multiple incrementally coded layers to improve video quality cumulatively. Each receiver interested in receiving the video subscribes to the base layer and to additional cumulative layers according to their available bandwidth.

In this paper we propose a novel adaptation of the multicasting scheme to transmitting light field video data. The proposed system multicasts the light field video data to the receivers where each receiver dynamically determines the parts of the IBR representation necessary to render their current viewpoint and subscribes only to the corresponding layers. In addition future viewing positions are predicted using Kalman filters and necessary streams are prefetched to prevent starving of the viewers.

In the remainder of this paper we first discuss the assumptions we made and describe details of the proposed system in Section 2. In Section 3 we present results of our viewpoint prediction system in addition to our simulations on the position dependent nature of the light field rendering. Finally we discuss our conclusions in Section 4.

## 2. SYSTEM DESCRIPTION

The proposed system uses light field video as its IBR representation. Light field video is an extension of the light field idea, where each frame is a separate light field or equivalently each view is a separate ordinary video stream. For the rest of this text we will assume a light field parameterization with two parallel planes, camera (*s-t*) plane and image (*u-v*) plane; however, discussions in this work can be applied to spherical light fields as well. In our multicasting framework there is one central multicast server with the complete light field video, which might be in compressed or uncompressed format, and several receivers requesting various views from the central light field video server. Only requirement we make on the light field video coding is that the streams from the cameras used to capture the light field video should be somehow independently available on server side. If an uncompressed light field video is available, streams from cameras can be compressed separately before being multicast. If the available light field video is compressed using a technique, which creates dependencies between camera views, then original camera streams may be reconstructed and multicast independently.

Current light field compression methods create dependencies between camera views. Some cameras are intra coded and views from other cameras are coded with reference to the intra-coded cameras. This is a very efficient way to compress light fields, which characteristically tend to demonstrate very coherent features. However in our proposed multicast system these dependencies have an adverse effect on the interactive nature of the system. In a multicasting scenario using compressed light fields at any given time instant the viewpoint of the viewer might change such that some of the new requested camera streams happen to be hierarchically dependent on several other streams, which are not required to actually render the view in question. However because of the dependencies, the viewer would have to subscribe to required streams and to all other streams, on which the required streams are dependent, causing overhead in network traffic. In the worst case when the view to be rendered requires a few streams, none of which are independently coded, the viewer might end up having to subscribe to many independently coded streams to reconstruct the view in question. We will leave a detailed analysis of this problem beyond the scope of this paper and for the purposes of this paper assume that the streams are independently available.

### 2.1 Receiver-driven Light Field Video Multicasting

In the receiver-driven layered multicasting framework as introduced in [8] the receivers are responsible for the decision on selecting the layers to join. This decision is normally based on the available bandwidth: receivers try to join to more layers as bandwidth becomes available and drop layers when network congestion occurs. We propose extending this framework by changing the decision criteria at receivers. Light field video is not coded as base a layer and incremental layers in a single dimension. The layers are a 2-D array of streams reflecting the physical arrangement of the cameras used to capture the light field video. Each receiver selects a set of layers to join from this 2-D array according to its current viewpoint. Clearly the available bandwidth sets an upper bound on the number of layers a receiver can simultaneously subscribe to. Therefore the receiver should constrain the user from going into regions where the total bandwidth of the required layers is greater than the estimated available bandwidth. In the case of a dedicated connection for the streaming of the light field video, the bounding region of the user can be predetermined by the bandwidth of the dedicated link. In the case where the link is shared with other applications the available bandwidth fluctuates and the bounding region will have to change dynamically. In the proposed system we will use the bandwidth estimation method described in [9]. Even though the packet-pair multicasting implementation requires that the routers in the network implement fair queuing (FQ), it converges much faster that using the packet loss as an indicator of available bandwidth as in [8].

### 2.2. Determining Required Streams

Given the light field coordinates of a ray, *(u,v,s,t),* the camera coordinates of the corresponding pixel, (*x,y*), can be determined by using a projective transform. Conversely given the viewpoint of the camera and the coordinates of the corners of the camera frame, the inverse of the projective transform can be used to determine corners of the corresponding quadrilateral, *Q*, in the camera plane. Once *Q* is determined then the required streams are a function of the interpolation method used [4]. For nearest neighbor interpolation the streams from cameras whose Voronoi cells in the camera plane intersect with the quadrilateral *Q* are required. If linear interpolation is used in the camera plane then the neighboring streams surrounding those needed for the nearest neighbor interpolation are required as well. Therefore the interpolation quality is an important factor in determining the required bandwidth.

### 2.3. View Prediction

As the viewpoint of the user changes the required streams might change. Due to network lags it is necessary to anticipate ahead of time when a new stream will be needed to render the user's viewpoint. Assuming there is no chance in the focal length (zooming) of the virtual camera, a viewpoint is determined by six variables, the position in 3-D space (*x, y, z*) and the Euler Angles ($\phi$, $\theta$, $\psi$). The proposed system uses six separate Kalman filters based on the model described in [10] to predict values of the variables in the next time instance, using the current position, velocity and acceleration information. The physical model, where acceleration is assumed to be partially linear, is shown below:

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ \ddot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} + \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix} w_k$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix}$$

Where $x_k, \dot{x}_k, \ddot{x}_k$ are the position, velocity and acceleration of one of the six variables in *k*-th sample, $w_k$ is the change in acceleration modeled as white noise during time interval *T*.

### 2.4. Joining and Leaving Streams

Joining a stream of the light field video involves a delay which is a product of two independent delays: join latency associated with the multicast network and the wait for an I-frame before the stream can be decoded. According to the analysis of the multicast join delay in [11], on a low-load multicast network the join latency can be found to be approximately linearly proportional with the number of hosts in the network at 0.59msec per host. At this rate the network latency equals the interval between frames

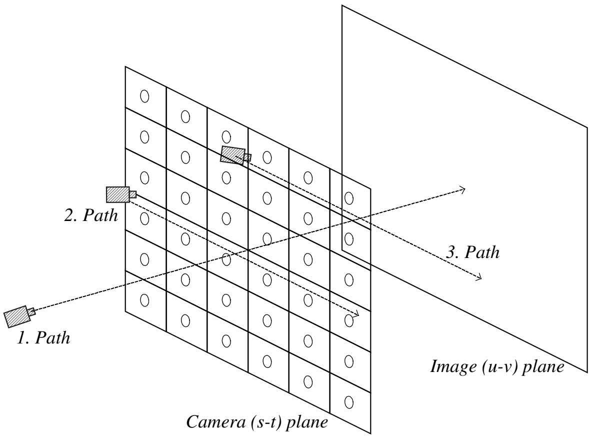

*Figure 1:Our hypothetical light field geometry with three virtual camera paths.*

when there are about 50 receivers in the multicast network. To be able to multicast light field videos to more users viewpoint prediction for more than one frame is necessary.

Even though the streams are independently coded from each other, we still assume compressed streams. When a viewer subscribes to a stream, it needs an independently coded frame before the stream is useful for rendering of the new virtual view. Since I-frames occur relatively infrequently in coded video, joining a stream would involve a long wait for an I-frame before the stream can be decoded and utilized to render a novel view. One way to counter this problem is inserting frequent I-frames into the camera streams but that would reduce the coding efficiency and increase the number of transmitted bits. A better solution to this problem is the incorporation of SP/SI-frames into the coded streams. SP/SI-frames are a feature of H.264 standard. SI-frames are encoded into the stream alongside the SP frames to provide error resiliency and random access. When a receiver joins a stream it first receives the corresponding SI-frame instead of the first SP-frame, SP-frames and B-frames, if there are any, are streamed normally afterwards. In [12] Karczewicz and Kurceren show that a properly encoded SP frame is only marginally larger than a regular P-frame. On the other hand the coding efficiency of an SI-frame is less than an I-frame. However this is not of much concern since an SI-frame is transmitted only once when the streaming starts. Therefore by replacing all P-frames in the stream with SP-frames and corresponding SI-frames, it is possible to allow random access to the stream a marginal increase in transmitted bits, but the number of stored bits at the server side increases substantially.

## 3. RESULTS

We have simulated viewing of light field video by using an hypothetical light field with 16x16 cameras in the camera plane which is 50cm x 50cm in size. The image plane is 50cm from the camera plane and each stream has a resolution of 512x512 pixels and a frame rate of 30 frames per second. An imaginary camera with a 640x480 pixel resolution and 50mm focal length was simulated along three paths. The first path starts 1.0m away from the light field image plane and moves in to 0.1m. It crosses the light field camera plane at (0.25,0.25,0.5) and the resulting Figure 2. clearly shows that much fewer streams are required when the imaginary camera is close to the light field camera plane. The second path is slightly behind the camera plane of the light field and traverses along positive x direction from 0.0m to 0.5m at 0.25m. During the traversal the camera points to point at the center of the light field image plane (0.25,0.25,0.0). The number of required streams is almost constant along this path, except for slight increases at the ends of the path where the pan becomes large enough and where the imaginary camera is close to the midpoints between actual cameras. The same traversal is repeated when the camera is positioned between the camera and image planes of the light field at a depth of 0.25m. This path requires much more streams first because it's further away from the camera plane and second a greater pan is required to keep camera pointed at the (0.25,0.25,0.0) point. The comparison of number of streams required for these two paths can be seen in Figure 3.

We implemented six separate Kalman filters to predict six parameters of the viewpoint. First the viewpoint in the next frame was predicted. Then the required cameras for the predicted viewpoint are found and compared with the required cameras for the actual viewpoints. The mean errors for the Kalman filters are around 3% percent for all viewpoint coordinates except the x-coordinate for which the error of the corresponding Kalman filter averages 17.4%. This relatively high error ratio for the x-coordinate is due to some abrupt changes in our viewpoint data, where the Kalman filter fails to predict the fast change. As a result of prediction errors the required streams do not perfectly match with the required streams for the actual measured viewpoints. Figure 4. shows that the number of missed streams for each frame is affected quite adversely from the prediction errors.

## 4. CONCLUSIONS

We have shown how the number of streams required for novel view rendering is dependent on the viewpoint defined by the coordinates and the orientation of the virtual camera. When the virtual camera is near the camera plane of the light field, the number of required streams is quite low and the light field video can be watched even over relatively low bandwidth connections, however as the virtual camera gets further away from the camera plane the number of required streams appears to be increasing roughly with the square of the distance.

Correctly requesting streams which will be required depends very closely on the performance of the viewpoint prediction. Therefore the prediction distance must be kept as low as possible to guarantee best possible prediction. Since there is little which can be done for the intrinsic multicast network join-latency, it must be ensured that there is as short a wait as possible for the decodable frame. Which means that the camera streams must be coded using SP/SI-frames to provide random access at a relatively low cost.

According to the ratio of the number of B-frames to the number of SP-frames in the stream the prediction distance changes. If there are no B-frames in the stream the prediction distance can be set equal to the multicast join-latency. As the ratio increases the prediction distance must be increased by the interval between SP-frames to account for the wait before the first SP-frame occurs. This way it is made sure that the requested stream will be available and decodable at the receiver side when it is needed. Current results of our viewpoint prediction system suggest that better overall performance can be achieved by not using B-frames to improve prediction performance at the cost of coding efficiency.

## 5. ACKNOWLEDGEMENT

This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV.

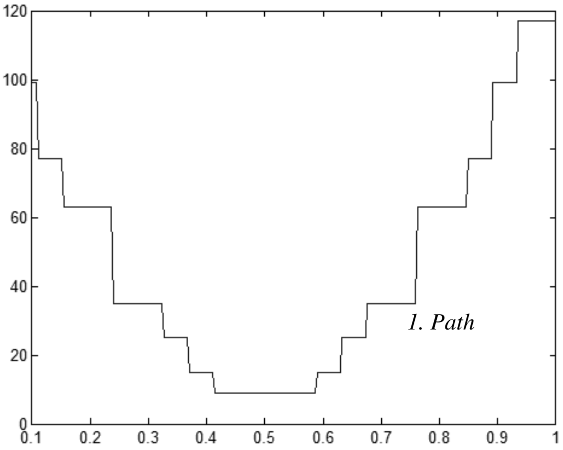

*Figure 2: As the camera moves along the Z-axis least number of streams are needed at the intersection with the light field camera plane*

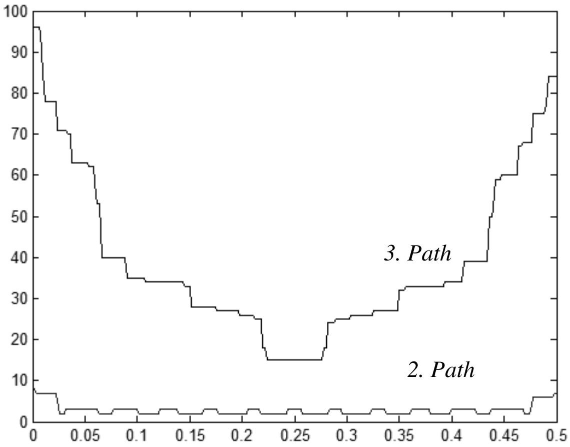

*Figure 3: As the camera moves parallel to the X-axis the number of required streams increases drastically if the camera is further away from the camera plane*

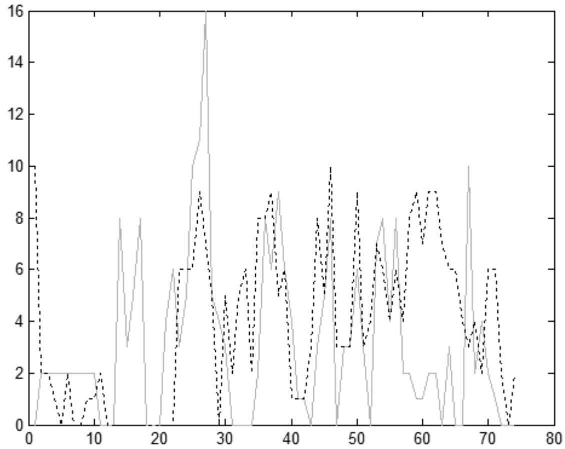

*Figure 4: Missed streams. The dashed curve is the number of actually required streams which were not predicted. The solid curve is the number of unrequired streams wrongly predicted.*

## 6. REFERENCES

[1] C. Zhang, T. Chen, "A survey on image based rendering – representation, sampling, and compression", Signal Processing: Image Communication 19(2004) 1-28

[2] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision", in *Computational Models of Visual Processing*, M. Landy, ve J. A. Movshon, Cambridge: MIT Press, 1991

[3] M. Levoy, P. Hanrahan, "Light Field Rendering", Proc. ACM Siggraph 1996, pp. 31-42

[4] S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, "The Lumigraph", Proc. ACM Siggraph 1996, pp. 43-54.

[5] I. Ihm, S. Park, R. K. Lee, "Rendering of Spherical Light Fields", Proc. The Fifth Pacific Conference on Computer Graphics and Applications, 1997. Pages:59 - 68

[6] X. Tong, R. M. Gray, "Interactive Rendering From Compressed Light Fields", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 11, Nov. 2003

[7] M. Magnor, B. Girod, "Data Compression for Light Field Rendering", *IEEE Trans. On Circuits and Systems for Video Technology*, Vol. 10, No. 3, pp. 338.343, April 2000.

[8] S. McCanne, V. Jacobson, M. Vetterli, "Receiver-driven Layered Multicast", ACM Sigcomm 1996

[9] A. Legout and E. W. Biersack, "PLM: Fast Convergence for cumulative layered multicast transmission schemes", SIGMETRICS 2000 6/00

[10] A. Kiruluta, M. Eizenman and S. Pasupathy, "Predictive Head Movement Tracking Using a Kalman Filter", IEEE Trans. On Systems, Man and Cybernetics, Vol. 27, No. 2, 1997

[11] D. Estrin, M. Handley, A. Helmy, P. Huang, D. Thaler, "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing", Proc. IEEE INFOCOM 1999, pp. 1090-1098

[12] M. Karczewicz, R. Kurceren, "The SP- and SI-Frames Design for H.264/AVC", IEEE Trans. On Circ. And Sys. For Video Tech., Vol 13, No.7, July 2003