Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

# Interactive transport of multi-view videos for 3DTV applications*

KURUTEPE Engin, CIVANLAR M. Reha, TEKALP A. Murat
(*School of Engineering, Koç University, Istanbul 34450, Turkey*)
E-mail: {ekurutepe; rcivanlar; mtekalp}@ku.edu.tr
Received Nov. 25, 2005; revision accepted Feb. 27, 2006

**Abstract:** The authors propose a novel method for transporting multi-view videos that aims to keep the bandwidth requirements on both end-users and servers as low as possible. The method is based on application layer multicast, where each end point receives only a selected number of views required for rendering video from its current viewpoint at any given time. The set of selected videos changes in real time as the user's viewpoint changes because of head or eye movements. Techniques for reducing the black-outs during fast viewpoint changes were investigated. The performance of the approach was studied through network experiments.

**Key words:** 3DTV, Multi-view video, Application-layer multicast, Join-latency
**doi:**10.1631/jzus.2006.A0830 **Document code:** A **CLC number:** TN919.8

## INTRODUCTION

Projects such as three dimensional television (3DTV) and free-viewpoint television (FTV) aim to enable viewers to freely roam in reconstructed 3D scenes, and ultimately to achieve a technology similar to the famous "Holodeck" in the Star Trek series. While the Holodeck goal may be somewhat distant for now, some forms of interactive 3D entertainment can be in living rooms in the very near future.

The first products to offer interactive 3D will probably be based on a technology to show two 2D views such that left and right eyes of the viewer see from their respective viewpoints, creating an illusion of 3D. Various enabling technologies for this, such as polarizing filters and glasses, shutter glasses, and autostereoscopic displays have already reached a certain level of maturity. To provide interactivity, such a system needs to know from where the viewer is looking at the scene so that correct views are fed to the eyes. The user's viewpoint can be determined by various techniques ranging from explicit use of a mouse to a complex head and eye tracking system.

### Techniques for rendering scenes in 3D

Once the viewpoint is determined, there are two broad approaches on how to generate the corresponding views. 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 3D scene is highly dependent on the scene geometry and is usually very high (Levoy and Hanrahan, 1996). Moreover accurately capturing 3D geometry of real world objects is still an unsolved problem. The other general approach, Image Based Rendering (IBR), aims to generate novel views of the scene using captured images from a multitude of viepoints. The idea behind the IBR systems is the seven dimensional plenoptic function, which describes all potentially available optical information in a given region (Adelson and Bergen, 1991). Various IBR systems, such as light fields (Levoy and Hanrahan, 1996) or the Lumigraph (Gortler *et al*., 1996), are simplifications of this

* Project (No. 511568) supported by the European Commission within Framework Program 6 with the acronym 3DTV

plenoptic function. Pure IBR systems do not assume an explicit 3D model of the scene. However there is a continuum of image and geometry based representations and a trade-off between the amount of geometry information and number of necessary views for a good rendering (Kang *et al*., 2000).

Rendering novel views from an IBR representation generally requires much less computational resources than rendering views by texture mapping geometric models. However, the reconstruction quality of IBR systems, such as light fields, depends on the sampling density in the camera plane or the availability of the scene geometry. As a result IBR representations for a good reconstruction quality require a large number of cameras to capture the scene and generate enormous amounts of data. Previous research (Levoy and Hanrahan, 1996; Zhang and Chen, 2004; Tong and Gray, 2003) has shown that raw data for a single static light field can reach up to several hundred megabytes or even gigabytes for high-resolution examples. On the other hand light fields contain highly coherent data and can be compressed extensively. High compression rates, in the order of 500:1 and 1000:1, have been reported for static light fields (Magnor and Girod, 2000; Tong and Gray, 2003), but these high compression rates come at the cost of difficulty in interactive viewing due to the dependencies created between the light field images. There is ongoing 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 make it feasible to stream them over domestic broadband connections in the near future.

There are other IBR systems, such as multi-view videos, which use the geometry information to reduce the required amount of data to achieve high quality rendering. The most common form of geometry information used in such systems is depth or disparity maps. Previous work (Fehn *et al*., 2004) showed that the depth information can be compressed to about 10%~20% of the video stream without perceivable quality loss. In the case of a system with 8 views and 8 corresponding depth maps, this corresponds to about 9 to 10 times the original bit rate of a single-view video. Therefore there still is a need for novel networking schemes, which efficiently use the available bandwidth able to send the multi-view videos over existing broadband connections.

**Multicasting solutions**

Multicasting is an established method which aims to efficiently transport packets from one or more senders to interested receivers. Multicasting paradigm tries to prevent sending duplicate packets to multiple receivers in the network to save bandwidth. In the network-layer multicast, the sender sends every packet only once. These packets get duplicated at multicast-enabled routers as needed and are forwarded to other interested routers and hosts. Even though network-layer multicasting is theoretically the most efficient method to distribute information to a set of interested receivers, it is not widely deployed in practice because it requires replacement of existing routers with special multicast capable ones. Although most of the new routers are multicast capable, because of security and other operational concerns, network operators often keep multicast functionality of their routers disabled. Therefore, large parts of the Internet are cut off from network-layer multicast capability (Banerjee *et al*., 2002).

The alternative to duplicating and forwarding packets in the network is to shift that functionality to the end-hosts. In application-layer multicast, packet duplication, forwarding and management of distribution trees are all accomplished through software at end points. Therefore, it can be widely deployed very easily with little or no investment at all. Application-layer multicast is not as efficient as network-layer multicast in two aspects: some packets end up travelling through more hops than they would have to if they were sent with network-layer multicast, and some physical links have to carry some duplicate packets (Chu *et al*., 2001). Two important factors to be considered when assessing the quality of an application layer multicast application are the amount of overhead traffic needed to construct and maintain the distribution tree and the join latency, which is the delay between the first request a host sends to join the multicast distribution tree and the first data packet it receives.

Both network-layer and application-layer multicast can only improve efficiency on the server-side, by preventing or reducing duplicate packets in the network. It was shown (McCanne *et al*., 1996) that the multicasting principle can be further adapted to transmission of multimedia data by multicasting scalable multimedia data in multiple layers and giving the control over which layers to receive to the

end-points. Previously (Kurutepe *et al*., 2005), we applied this idea to dynamic light fields and proposed network-layer multicasting as a possible solution for the transport of dynamic light fields, where each view is treated as an independent multicast layer. In this paper we extend our previous work and propose a novel adaptation of the NICE (Banerjee *et al*., 2002) application-layer multicasting scheme to transmit IBR representations or multi-view videos in general.

In NICE protocol, members of a multicast group organize themselves into small geographically close clusters using ping roundtrip time measurements. These clusters also form the lowest layer, $L_0$, in a hierarchy. The most central member in each cluster is elected as the cluster leader and promoted to the next higher layer, $L_1$. Cluster forming, leader selection and promotion procedures are repeated recursively in higher layers such that leaders of clusters in $L_n$ become members of $L_{n+1}$ until a single member becomes the root of the hierarchy at the highest layer, $L_{n\max}$.

The data delivery paths of the overlay multicast network are implicitly defined by the hierarchy, eliminating the need to maintain states for delivery trees and control meshes separately. When a host $h$ receives a data packet from another host $p$, $h$ forwards the packet to all clusters on all layers where $h$ is a cluster member and $p$ is not.

Organizing the multicast group members in a hierarchy also has the benefit that each member only keeps detailed state about its closest neighbors. Therefore, the state information exchange packets are relatively few and thus control overhead is low when compared to other overlay multicast protocols such as NARADA (Chu *et al*., 2001).

In our proposed 3DTV system, independent overlay distribution trees are constructed for each camera view and for each depth map stream in the representation. Each receiver determines the parts of the IBR representation necessary to render their current viewpoint and subscribes only to the corresponding distribution trees. The future viewing positions are predicted using Kalman filters and necessary streams are prefetched to prevent black-outs during fast viewpoint changes.

The remainder of this paper is organized as follows. First the details of various parts of the proposed system are described in Section 2. The results are presented in Section 3. And, finally the conclusions and a discussion of future work are presented in Section 4 and Section 5.

## SYSTEM DESCRIPTION

The proposed system assumes that the IBR representation is in the form of multi-view video with optional depth or disparity information. Both the camera views and the depth information are in the form of compressed video streams. We use H.264 compression to independently encode both video streams and depth videos. Even though the multi-view video can be compressed more efficiently using more advanced multi-view coding (MVC) techniques, it is difficult to use these in systems as the one proposed in this paper which selectively transmits parts of the complete scene description. All MVC techniques exploit the spatial redundancy in multi-view videos to improve compression. This idea is very similar to motion compensation in video coding and it creates dependencies between camera views, similar to the dependencies created by motion compensation in single view video coding. These dependencies help to reduce the total size of the multi-view video by removing redundancies in the representation, but they also make random access into the multi-view video very difficult. If the proposed system were to be used with MVC, each inter-coded view requested by a client will also cause the corresponding intra-coded views, from which it was predicted, to be streamed to the viewer. In extreme cases this might cause many intra-coded views to be fetched for a single requested inter-view, making very inefficient use of the network resources. Therefore, network bandwidth can be saved by selectively transmitting independently coded video streams instead of a complete MVC compressed multi-view video stream at the cost of some disk storage at servers at least until a more flexible compression approach becomes available.

In the proposed multicasting framework there are one or more central multicast streaming servers, each with some streams corresponding to different views of the multi-view video data. In addition to the central streaming servers there are "professional" peers, which implement the NICE application-layer multicast protocol and form a dynamic distribution

network for multi-view video streams. End users run client software on their computers or set-top boxes which requesting streams from a known multicast peer and passing received stream to the rendering software. The rendering software is specific to the IBR technique used and must be thought of as a separate module from the multicast network. The rendering module has to accomplish two important tasks: to render the current view using the streams fetched by the multicast client and to track and predict the viewpoint of the user and to instruct the multicast client to request the relevant streams.

**Receiver-driven multi-view video application-layer multicasting**

In the receiver-driven layered multicasting framework as introduced by McCanne *et al*.(1996), the receivers are responsible for 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 criterion at receivers. Unlike multicasting of single viewpoint video where the join or leave decisions are made only according to the bandwidth limitations, the decision on which layers to join must be made according to two criteria in the case of multicasting multi-view video. Both the available bandwidth and the required streams play a role in the decision. The number of subscribed layers is clearly limited by the available bandwidth. Joining more layers than allowed by the available bandwidth will cause network congestion and prevent proper streaming of all layers. When the number of required layers is less than what the available bandwidth allows, the decision is trivially simple: all the required layers should be joined. However when the total data rate of all required layers exceeds the bandwidth limitation, there are two possibilities: The rendering module should either constrain the users from changing their viewpoints into forbidden regions where the data rate of all required streams is larger than the available bandwidth, or refrain from joining some of the required layers in those regions at the cost of rendering quality. The performance of the second approach is very closely related to the chosen IBR representation and must be implemented in conjunction with the rendering module.

The determination of the streams required for rendering of the current viewpoint depends on the IBR representation employed in the multicasting system. We previously studied how the set of required views changes for a dynamic light field as the viewer moves its head on a predetermined trajectory and reported a straightforward method to determine which views are required to render the current viewpoint (Kurutepe *et al*., 2005). Although we leave a detailed analysis of this question for other multi-view video representations out of the scope of this paper, we expect the determination of the required streams to be a relatively trivial problem given the viewpoint of the user.

**Viewpoint prediction**

As the viewpoint of the user changes the required streams might change. Even though zooming can be artificially implemented in display systems, the human visual system lacks this facility. Therefore for a natural 3D experience it can be safely assumed that the focal length of the virtual camera at the viewpoint does not change. In that case a viewpoint is determined by six variables, the position of the virtual camera in 3D space ($x$, $y$, $z$) and the Euler angles relative to the world coordinate system ($\phi, \theta, \psi$), which define the extrinsic parameters of the virtual camera.

The proposed system uses six separate Kalman filters based on a partially linear acceration model (Kiruluta *et al*., 1997) to predict values of the variables in the next time instance, using the current position, velocity and acceleration information. The physical model is shown below:

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

$$y_k = [1 \quad 0 \quad 0][x_k \quad \dot{x}_k \quad \ddot{x}_k]^{\mathrm{T}},$$

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 modelled as white noise during time interval $T$, which corresponds to the interval between frames.

The viewpoint prediction must be optimized in relation to two parameters: the join latency and the decoding delay after the successful join operation

during which the received stream is not yet decodable.

The entire join procedure in NICE protocol requires $O(\log N)$ packet exchanges, where $N$ is the number of members in the multicast group. Thus the join latency can be estimated as $O(\log N)$ times the average RTT in the multicast group. On the other hand, while the join procedure is in progress, the newly joining member is temporarily added to the data path of the last contacted cluster leader. Therefore the new members begin receiving data packets before they have completely joined the hierarchy and according to (Banerjee, 2002), they should experience a short join latency after which they receive the first frame.

However the first received frame is not necessarily decodable if it happens to be a P- or B-frame. Since I-frames occur relatively infrequently, 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 using only I-frames. However that would reduce the coding efficiency and substantially increase the number of transmitted bits.

Clearly there is a trade-off between the viewpoint prediction performance and the number of predicted P- and B-frames between periodic I-frames. As the number of the predicted frames increases, the viewpoint prediction distance must be increased as well, reducing the accuracy of viewpoint prediction. Depending on the nature of the employed IBR representation, errors in viewpoint prediction will have varying effects on the performance of the whole system. As previously shown (Kurutepe *et al*., 2005), when the light fields are used as the IBR representation, the effects of the prediction errors are quite significant. For that example, it might be beneficial to completely forego the coding efficiency provided by the predicted frames in favor of better reconstruction quality.

The proposed system uses multi-view videos as the IBR representation. The most important difference between multi-view videos and other IBR representations without some geometry information, such as light fields, is that the distance between views is much larger. Due to this greater base-line between cameras, new stream requests can only happen for more significant viewpoint changes. Therefore the prediction errors are less serious in the proposed system. As long as the errors are not big enough to cross the boundary where switching from one view to another occurs, the rendering result will not be affected. As a result, the prediction distance can be increased and larger prediction errors can be tolerated. This permits less frequent I-frames in the bit streams and improves coding efficiency.

**Overlay distribution tree management**

An independent NICE hierarchy is maintained for each view in the multi-view video distribution. Each overlay multicast peer joins and leaves these hierarchies as it needs the corresponding view to render its current viewpoint. Each client has a list of geographically close overlay multicast peers. When a viewer requires a certain stream to render the current view, its client contacts one of the known peers, which is estimated to have the best possible connection to the client and requests for the stream. If the peer in question is already a member of the corresponding hierarchy, it already has the requested stream and starts forwarding packets to the new client immediately. If, however, it is not a member of that multicast group, according to the NICE protocol it contacts the bootstrap node and requests to join the multicast group. The bootstrap node returns with the address of the cluster leader in the topmost layer of the hierarchy. The newly joining peer contacts the topmost cluster leader to obtain the addresses of its children in the next lower layer, pings those peers, and contacts the closest one to obtain the addresses of its children. This procedure repeats until the new peer has joined a cluster in the lowest layer.

## RESULTS

The NICE protocol has been modified such that hosts can simultaneously be members of multiple parallel multicast hierarchies. Network tests have been carried out in the Koc University network. One computer was assigned as the bootstrap node for all multicast hierarchies, and another computer served all streams of the multi-view video, which was generated using H.264 compression from 8 camera views of the "MPI Kung-Fu Girl" synthetic multi-camera sequence. In this dataset each camera has 320×240 pixels and 200 frames. The resulting video was

compressed with I-B-B-P-B-B-I-… frame sequence at about 240 kbps per view with 5 predicted frames between two I-frames. Ten other computers were setup to simulate join requests of viewers of the multi-view video by joining and leaving streams completely at random.

Join latency is defined as the duration between the join request and the first data packet received from the multicast hierarchy. Network measurements showed that the join latency of the proposed system is low. Fig.1 shows a histogram of the results of the join latency experiments. As can be seen, the new member almost always starts receiving data packets in less than 40 ms except for a few outliers. The median of the data is about 25 ms and the mean is just above 42 ms. Even though the whole join procedure depends on the size of the multicast group and takes $O(\log N)$ message exchanges, the new member becomes a part of the data delivery tree as soon as it first contacts the topmost cluster leader. Therefore the join latency is typically independent of the group size.

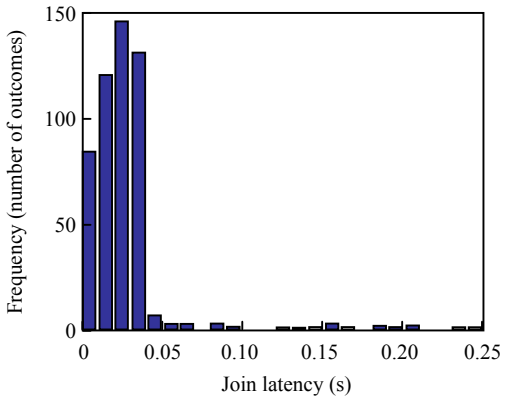


**Fig.1 Histogram of join latencies**

We have measured the error of the Kalman filter viewpoint prediction subsystem at various prediction distances, $d$, which is given in number of frames, i.e. when $d$=5 the viewpoint predictor tries to predict where the viewer's head will be after 5 frames. Unsurprisingly, the prediction errors increase with $d$. Fig.2 shows how the prediction error changes during a sequence at various prediction distances. If the error is large enough to cause a wrong stream to be requested, it will adversely affect the rendering quality of the system.

In order to find out how severely the prediction

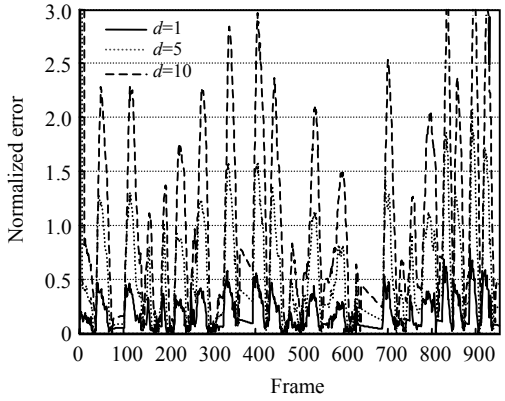


**Fig.2 Viewpoint prediction error where $d$ is the prediction distance**

errors would affect the viewing experience, we have also studied the correlation between the errors and missed streams. An autostereoscopic display system with a head tracking camera was considered. The display was connected to a stereo renderer, which generates two novel views from a 7-camera multi-view video according to the viewer's head position. The viewpoint prediction was run on the recorded head data of someone moving back and forth in front of the display while viewing the multi-view video. Table 1 shows the performance of several viewpoint prediction distances on the test system. Predictably, the percentages of the missed and redundantly transmitted streams increase with the prediction distance. When the prediction distance is set to 5, which also corresponds to the number of predicted frames between the intra frames of our multi-view video, 4.14% of the streams could not be predicted accurately and would not arrive on time to the client. However, the predictor only might miss a stream if the renderer suddenly requires a neighboring view in addition to the currently available views. If the head movement, which initially caused the stream switch, is continuous the predictor catches up usually within one or two frames. Therefore during fast head movements near the boundaries where another stream is needed the renderer might starve for a single stream for a few frames, resulting in poorer 3D rendering of the scene for a short time. Fortunately this kind of errors does not cause a complete black-out of the multi-view video, since during the time when the new view is not available the other views are still available and utilized for 3D rendering.

**Table 1 Percentage of missed and redundantly transmitted streams as a function of viewpoint prediction distance (*d*)**

| *d* | Missed (%) | Redundant (%) |
|---|---|---|
| 1 | 3.30 | 3.67 |
| 5 | 4.14 | 4.51 |
| 10 | 6.60 | 6.97 |
| 20 | 13.48 | 10.32 |
| 30 | 17.57 | 12.32 |
| 50 | 21.43 | 16.09 |

## CONCLUSION

The main contribution of our work described in this paper is selectively transmitting required parts of the multi-view video representation to viewers through parallel and independent application layer multicast delivery trees.

The mostly constant join latency measurements compare very well with the join latency performance of network-layer multicast protocols, which typically have about 0.59 ms latency per peer in the multicast group (Estrin *et al*., 1999) under low load network conditions. Thus, as the group sizes increase to about 40 members, our parallel overlay multicast using the NICE protocol should start to perform better than network-layer multicast.

Since the network delays are found to be relatively low the main source of delay in the proposed system is the wait for an I-frame after the peer joins the multicast group. The frequency of I-frames represents an important trade-off which affects the performance of the whole system: prediction distances can be kept low by using more frequent I-frames; however that comes at the cost of coding efficiency. On the other hand, when fewer I-frames are used, the prediction distance needs to be increased to ensure that the stream is available and decodable at the viewer by the time it is needed for reconstruction. Although this approach improves the coding efficiency of the streams, large prediction errors associated with longer prediction distances might cause wrong streams to be fetched and adversely affect the rendering quality.

## FUTURE WORK

The performance of the proposed system will be investigated on real world wide area networks and results will be reported.

## ACKNOWLEDGEMENT

The authors sincerely thank Prof. Bobby Bhattacharjee and Seungjoon Lee for supplying the sources for the NICE protocol.

## References

Adelson, E.H., Bergen, J.R., 1991. The Plenoptic Function and the Elements of Early Vision. *In*: Landy, M., Movshon, J.A. (Eds.), Computational Models of Visual Processing. MIT Press, Cambridge.

Banerjee, S., Bhattacharjee, B., Kommareddy, C., 2002. Scalable Application Layer Multicast. Proc. ACM SIGCOMM.

Chu, Y.H., Rao, S.G., Zhang, H., 2001. A Case for End System Multicast. Proceedings of ACM SIGCOMM.

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

Fehn, C., Hopf, K., Quante, Q., 2004. Key Technologies for an Advanced 3D-TV System. Proceedings of SPIE Three-Dimensional TV, Video and Display III. Philadephia, PA, USA, p.66-80.

Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F., 1996. The Lumigraph. Proc. ACM Siggraph, p.43-54.

Kang, S.B., Szeliski, R., Anadan, P., 2000. The Geometry-Image Representation Tradeoff for Rendering. Proceedings of 2000 International Conference on Image Processing, **2**:13-16.

Kiruluta, A., Eizenman, M., Pasupathy, S., 1997. Predictive head movement tracking using a kalman filter. *IEEE Trans. on Systems, Man and Cybernetics*, **27**(2):326-331. [doi:10.1109/3477.558841]

Kurutepe, E., Civanlar, M.R., Tekalp, A.M., 2005. A Receiver-Driven Multicasting Framework for 3DTV Transmission. European Signal Processing Conference.

Levoy, M., Hanrahan, P., 1996. Light Field Rendering. Proc. ACM Siggraph, p.31-42.

Magnor, M., Girod, B., 2000. Data compression for light field rendering. *IEEE Trans. on Circuits and Systems for Video Technology*, **10**(3):338-343. [doi:10.1109/76.836278]

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

Tong, X., Gray, R.M., 2003. Interactive rendering from compressed light fields. *IEEE Trans. on Circuits and Systems for Video Technology*, **13**(11):1080-1091. [doi:10.1109/TCSVT.2003.817626]

Zhang, C., Chen, T., 2004. A survey on image based rendering—representation, sampling, and compression. *Signal Processing: Image Communication*, **19**(1):1-28. [doi:10.1016/j.image.2003.07.001]