# CLIENT-DRIVEN STREAMING OF MULTI-VIEW VIDEOS FOR INTERACTIVE 3DTV

by

Engin Kurutepe

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical & Computer Engineering

Koç University

September, 2006

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Engin Kurutepe

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. A. Murat Tekalp

Prof. M. Reha Civanlar

Assist. Prof. Yücel Yemez

Assist. Prof. Oğuz Sunay

Dr. Aljoscha Smolic

Date: _____

*To my parents and to my beloved one*
*for their endurance*

# ABSTRACT

We present a novel 3-D multi-view video delivery scheme that allows users watch 3-D content interactively with significantly reduced bandwidth requirements by transmission of small number of views selected according to viewer's head position. The proposed scheme can be used for point-to-point delivery to a single user and/or multicast delivery to many users. Furthermore, the scheme can transport dense light-field-like multi-view sequences or multi-view sequences together with depth information.

In the proposed scheme, the user's head motion is tracked to select the required views dynamically. The system then allocates more bandwidth to these required views to render the user's current viewing angle. In order to reduce the adverse effects of delays due to network and stream switching, required view points are predicted and pre-fetched. Highly compressed lower quality views are also sent alongside the selected high quality views to provide protection against stoppage due to erroneous viewpoint predictions. A novel metric based on the abruptness of the head movements and delays in the system is introduced to help in determining the number of the additional lower quality views to be transmitted. The proposed system makes use of Multi-View Coding (MVC) and Scalable Video Coding (SVC) concepts together to obtain improved compression efficiency while providing flexibility in bandwidth allocation to the selected views. Rate-distortion performance of the proposed system is reported under different conditions. Use of the introduced metric in determining the optimal operating point, i.e., the number of lower quality views that need to be transported, is demonstrated.

Overlay multicast is employed to reduce the bandwidth requirements when the proposed scheme is used for transporting 3-D content to several users. An application-layer multicast protocol, called NICE, was modified for this purpose. NICE is chosen

due to its lower control overhead and short join-latency when compared to other application-layer multicast protocols; reducing the network delay for the proposed selective multi-view video delivery system.

# ÖZETÇE

Seyircinin konumuna göre seçilmis birkaç bakışı yollayarak kullanıcıların 3-B materyalleri etkileşimli ve düşük bant genişliği ile izlemesini sağlayacak yeni bir 3-B çok bakışlı video iletimi yöntemi sunuyoruz. Önerilen yöntem, hem noktadan noktaya tek alıcıya gönderim için hem de çok kullanıcıya çoğagönderim ile çalışabilir. Dahası sistemimiz hem daha yoğun ışık alanı türünde çok bakışlı videolarin hem de derinlik bilgisi ile daha seyrek çok bakışlı videolarin iletiminde kullanılabilir.

Önerilen yöntemde gerekli bakışlari seçmek icin kullanıcının kafa konumu sürekli olarak takip edilir. Kullanıcının göreceği görüntüyü yüksek kalitede geriçatabilmek için sistem seçilen bakışlara daha fazla bant genişliği ayırır. Ağdaki gecikmelerinin ve akış değişimlerinin istenmeyen etkilerini azaltmak için kullanıcının bakışı önceden tahmin edilir ve ihtiyaç duyulan akışlar önceden istenir. Buna ek olarak, tahmin hatalarından kaynaklanabilecek duraksamaları engellemek için yüksek derecede sıkıstırılmış komşu bakışlar da ihtiyaç duyulan bakışlara ek olarak yollanır. Yollanacak komşu bakış sayısını belirlemek icin kafa hareketlerinin aniliğini dikkate alan yeni bir metrik kullanılmıştır. Önerilen sistem, Çok Bakışlı Video Sıkıştırma (MVC) ve Ölçeklendirilebilir Video Sıkıştırma (SVC) fikirlerinden faydalanarak hem daha verimli sıkıştırma hem de daha esnek bakış seçimi sağlamayı amaçlar.

Çok bakışlı videolar önerilen yöntemle çok sayıda kullanıcıya gönderileceği zaman, uygulama katmanı çoğagonderim kullanılır. NICE olarak adlandırılan bir uygulama katmanı çoğagonderim protokolu, önerilen sisteme uyarlanmıştır.

# ACKNOWLEDGMENTS

First, I would like to thank my supervisors Prof. A. Murat Tekalp and Prof. M. Reha Civanlar, who have been a great source of inspiration and provided the right balance of suggestions, criticism, and freedom.

I am grateful to members of my thesis committee for critical reading of this thesis and for their valuable comments.

I would also like to thank to Dr. Bobby Bhattacharjee and Seungjoon Lee from University of Maryland for providing the sources for the NICE protocol and their helpfulness in answering my questions. Philipp Merkle from HHI must also be acknowledged here for answering my JSVM related questions with endless patience. And finally, Christoph Fehn from HHI, for our stimulating discussions and his valuable comments during my visit in Berlin.

Naturally, my office-mates, especially M. Emre Sargin, Ulas Bagci and I. Ekin Akkus, more than deserve to be mentioned here, for their friendship, for being always there to help and for acting as a sounding board for new ideas.

Last, but definitely not the least, I thank my family and my dearest, Burçin Behram, for giving me their love and support when I most needed it and for tolerating me at my worst.

# TABLE OF CONTENTS

# LIST OF FIGURES

# NOMENCLATURE

| | |
|---|---|
| 3DTV | Three Dimensional Television |
| FTV | Free Viewpoint Television |
| MVC | Multi-View Coding |
| SVC | Scalable Video Coding |
| MVV | Multi-View Video |
| RTT | Round Trip Time |
| GOP | Group Of Pictures |
| ALM | Application Layer Multicast |
| GMM | Gaussian Mixture Model |

# Chapter 1

# INTRODUCTION

The domestic entertainment technology is changing. The transition to HDTV is almost complete in the USA and Japan, and it is making itself felt in Europe. Many content producers have already switched to HD. In the mean time there is a growing research interest worldwide in the next logical step: 3DTV and FTV. While very similar in concept, and mostly used interchangeably, there is a difference in definition between the two terms: 3DTV systems aim to create the depth perception, whereas FTV refers to systems which allow interactive selection of viewpoint [1].

Both 3DTV and FTV systems require numerous enabling technologies in order to reach a certain maturity level. These enabling technologies cover a wide area of active research including but not limited to 3D scene acquisition, 3D scene representation, compression, transmission, multi-dimensional signal processing and 3D displays. All of these fields have seen tremendous advances in recent years, thanks to growing global research interest and international and national funding efforts such as European ATTEST and 3DTV NoE's, Japanese FTV project and Tele-Immersion project in the USA.

3DTV transmission, when compared to other areas listed above, received relatively little research effort, mostly due to the lack of a feasible compressed representation which can be transmitted over realistic networks. Geometry based computer graphics representations do not provide photographic quality levels and real-time rendering at the same time, and various image-based representations were much to big to be streamed to households over domestic connections. However, due to recent successful research on 3D scene representation and compression technologies, we now have effi-

cient and concrete compressed representations, such as multi-view videos compressed using MVC, to transmit.

However, even with MVC compression, multi-view videos are quite large. This represents two problems for an IP-based 3DTV streaming system. First, the viewers need a fast connection in order to watch 3DTV at a good quality and second, 3DTV servers need tremendous amounts of bandwidth in order to serve numerous clients. Both of these limitations can be overcome with the help of more intelligent transmission schemes. In this thesis we present such a 3DTV transmission scheme with a scalable encoding of Multi-View Videos, which makes uses of MVC and SVC concepts, and gives clients control about which parts of the MVV they need to receive. This selective delivery approach is aimed at reducing the bandwidth requirements and is the topic of Chapter 2. In order to address the bandwidth requirements at the server, in Chapter 3, we propose to extend the system detailed in Chapter 2 with application-layer multicasting, such that the server only needs to transmit to several peers, which in turn forward data to other peers, building an overlay delivery tree.

# Chapter 2

# CLIENT-DRIVEN SELECTIVE DELIVERY OF MULTI-VIEW VIDEO

## *2.1 Introduction*

Although dynamic holography is the ultimate goal in 3-D video and TV, early systems create 3-D viewing experience via stereoscopy by showing a scene from slightly different angles to the left and right eyes of a viewer. Various methods can be employed in order to generate these views. The most straight-forward approach is 3-D warping of an image using the associated depth map. This approach has been intensively studied in the literature, e.g., in the European ATTEST project, and it has been reported that the depth map can be compressed to about 10-20% of the video stream [2]. There is an ongoing MPEG standardization effort for the transport of a video-plus-depth representation [1]. Even though this approach is simple and cost efficient to implement, it offers limited flexibility, since the rendering quality quickly deteriorates due to disocclusions and discontinuities in depths, as the viewer moves away from the original camera angle [3].

The N-view-plus-M-depth representation has been proposed as a promising solution to address the limitations of the video-plus-depth representation, and MPEG has recently established an ad-hoc group to study challenges and opportunities offered by this representation [4]. When rendering from this representation, unlike the single-view-plus-depth case, the holes due to depth discontinuities can be filled using pixels from neighboring views. The most straight-forward method to generate views using N-views and M-depths is to 3-D warp the nearest two views to the desired view and intelligently blend them such that the holes are filled. Due to imperfections in the

depth maps, this method can leave ghost-like shadows near the object boundaries, which can be removed by using more sophisticated rendering algorithms [5], which make use of additional boundary masks. The boundary maps utilized in [5] are very sparse and can be transmitted without much overhead.

Another alternative is using N-view sequences without depth maps, which is called the light field representation when N is large [6]. Elimination of the need for depth determination is a significant advantage of this approach. Instead of generating novel views using depth information, two views closest to the viewer's eyes are selected and displayed when such a representation is used. [7] demonstrates such a system developed in MERL Labs, where views from 16 cameras are projected onto an autostereoscopic screen. As the user moves through the viewing space, views from the appropriate cameras are directed to the eyes with the help of the lenticular lenses on the screen.

With or without depth information, multi-view representations require large amounts of data, but the correlations between views can be exploited for compression. Multi-view coding (MVC) is an active area in the JVT and the standardization effort is in progress. [8] describes the current state of the art and reports significant compression gains over simulcast coding where each view is compressed independently. However, even with the MVC, bit-rates for multi-view streams are very high: 38dB PSNR at about 5 Mbps is a common operating point for a 640x480, 30fps, 8 camera sequence with MVC encoding. Moreover, previous research on single-view-plus-depth sequences [2] suggests that with the addition of depth maps and other auxiliary information such as boundary mask, the bandwidth requirements could increase as much as 20%, which renders 3-D TV service over current domestic Internet connections practically impossible.

In order to reduce the transmission bit-rate of multi-view sequences we observe that for a single-user, only two views are sufficient at any given time to create 3-D perception. Therefore, tracking users' head and selectively transmitting only the required views from the current viewing angle of the user can save significant band-

width [9] [10]. Since the transmitted views will vary in time according to user head position in free-view 3D video, we need random access to all views in the bitstream; This requirement cannot be met by MVC because of its complex dependency structure, unless all views are transmitted. When the views are simulcast coded, random access into each stream can be achieved at the cost of reduced compression efficiency; however, this cost is usually more than offset by the reduced number of transmitted views. On the other hand, this approach is highly sensitive to the delay between the request for the stream and its display. Therefore, in addition to tracking of user's head position, the future positions need to be predicted as well.

There is a wealth of previous research on predictive head tracking for 3-D display systems. In [11], Azuma and Bishop formulate a theoretical frequency domain framework in order to evaluate the performance of prediction algorithms and demonstrate on their experimental setup that predictive Kalman filters are suitable for 3-D applications. In [12], Wu compares simple extrapolation, Kalman filtering and grey system based predictive tracking algorithms and reports that both Kalman and grey system based algorithms achieve similar error performance for 3-D applications. In light of these results we have decided to use predictive Kalman filtering for our prediction system as described in [13] due to its low computational cost and ease of implementation. Not surprisingly, selective delivery systems perform poorly when the prediction fails (during sudden random movements etc.) and wrong views are transmitted, which do not correspond to the user's actual position at their play-out time.

Our motivation in this chapter is to significantly reduce the bit-rate requirements for real-time interactive transmission of multi-view sequences (with or without additional depth information) to enable domestic 3-D TV services. To this end, we propose to deliver low quality neighboring views in addition to the high quality views, which are actually required. The low quality views are delivered in the form of a base layer encoded using MVC and the high quality views are obtained using specially encoded enhancement layers. The number of the views contained in the base layer MVC can be changed in order to adapt to the actual operating conditions and the bit-rate al-

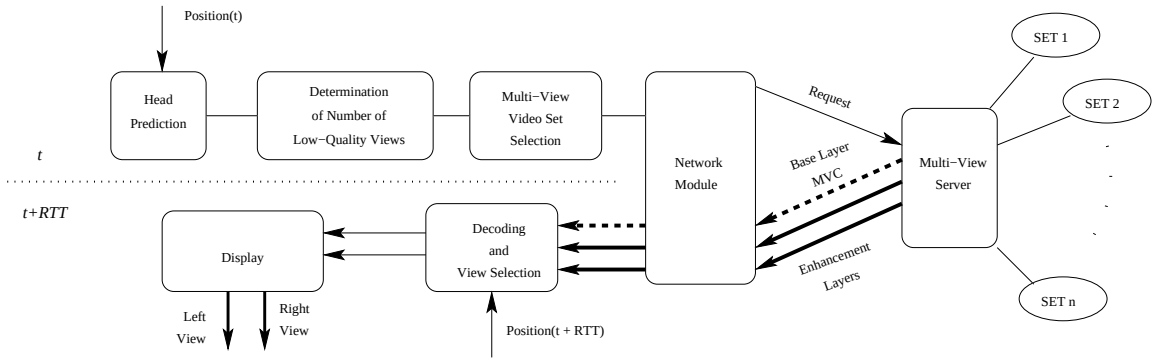



Figure 2.1: Overview of the delivery system with two low bit-rate side-streams.

location between the base and the enhancement layers can be determined for optimal system performance. This chapter is organized as follows: In Section 2.2 we describe the proposed system in detail. The results are presented in 2.3 and our conclusions are presented in 2.4.

## *2.2 System Description*

We propose to deliver a multi-view videosequence at two quality levels: $M$ views out of the total $N$ of the multi-view sequence are encoded using MVC at a lower bit-rate as a base layer. On top of this base layer, enhancement layers are encoded in order to improve the quality of selected views. The system first determines the user's current head position and a Kalman-filter based prediction subsystem predicts the user's head position $d$ frames into the future. Using the predicted positions, an error metric is computed at the receiver to determine the number of views to be included in the base layer, $M$. We assume that the bandwidth available to the user is limited, therefore as $M$ increases an increased proportion of the bandwidth is allocated to the base layer. This necessitates an intelligent allocation of bandwidth between the base layer MVC and the enhancement layer stream. The server has severals sets of the same multi-view video. Each set is encoded using a different $M$-number and has different bit rate allocation ratios between the base layer and the enhancement layers. The client switches to the appropriate set of streams according to the user's predicted

head position, its bandwidth and current $M$-number. The low bit rate base layer MVC allows the user to keep watching stereo video, albeit at a lower quality, when prediction fails until correct high quality streams arrive from the server. According to subjective quality tests reported in [14] and [15], as long as one of the eyes sees a high quality view, the lack of details in the low quality view seen by the other eye is masked by the human visual system. Therefore, as long as at least one of the views is delivered in high quality, the viewers might not even notice the low quality view in real-world implementations of the proposed selective transmission scheme, as long as the duration of such low quality periods are kept short. In this section, the server side issues are first described in Sections 2.2.1 and 2.2.2. Then the details of the receiver-side are described in the Sections 2.2.3 and 2.2.4.

### *2.2.1 MVC Base Layer with Simulcast Enhancement Layers*

The proposed approach to selectively transport the required views while providing interactivity, is using MVC coding with additional enhancement layers. This involves encoding all views together using MVC as a lower bit-rate base layer. The base layer is spatially downsampled to reduce the encoded video size. In addition to this base layer, an enhancement layer is generated for each view as follows: first the decoded video for each view in the MVC stream is upsampled to the full resolution of the original video and then the difference between the original videos and the decoded MVC videos are encoded using AVC/H.264. Alternatively the spatial scalability features of the emerging SVC standard can be used for this purpose in a similar fashion. These enhancement layers are independently coded to provide greater flexibility in switching from one view to the next. The coding structure of the proposed MVC plus enhancement layers can be seen in Fig. 2.2. This proposed structure benefits from the coding gain offered by MVC, while providing significantly greater flexibility in view selection, such that a user receiving the base layer and the enhancement layer for view $n$, is able to see it at a high quality, while all other views would be available at a lower quality.

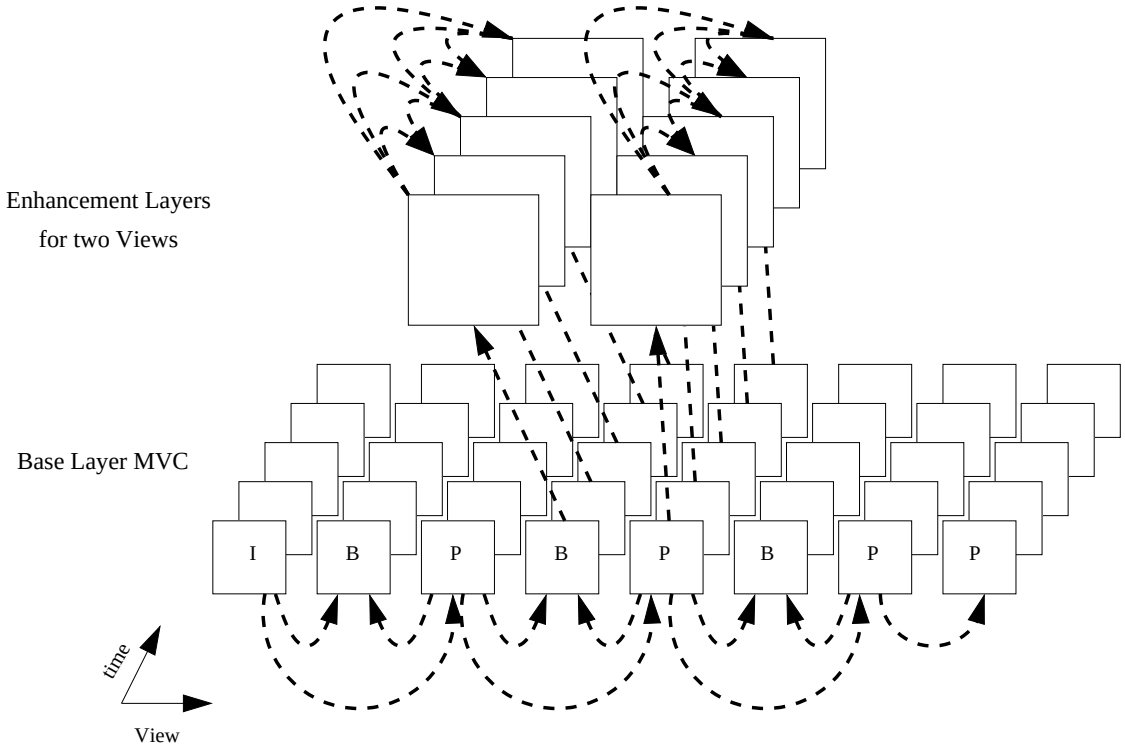



Figure 2.2: An example coding structure for the Base Layer MVC with enhancement Layers.

The multi-view video server has several previously encoded sets of base layer MVC plus enhancement layers. Each of these sets is a complete representation of the original multi-view sequence but the base layer MVC stream(s) in each set contains a different number of views. Whereas the base layer MVC of one set may have all views, another set may contain several overlapping base layer MVC streams such that each one has only four views. This allows the receiver to adapt itself to actual conditions and request the base layer MVC with as many views as dictated by the error statistics of its head position prediction system.

The proposed system has to handle three different sorts of stream switching. Adaptation of the number of views contained in the base layer MVC representation, $M$, and bit-rate allocation between base layer and enhancement layers is handled across GOPs, by switching from one multi-view video set to a different set at the start of a GOP, if necessary. Similarly, the base layer MVC and enhancement layers also need

to follow the user's head position. Both is achieved by switching the stream at the beginning of a GOP. It best to use a shorter GOP for the enhancement layers because they need to be able to follow the user's head position more closely, whereas a longer GOP may be utilized for the base layer MVC for more efficient compression.

In Section 2.3, we compare the proposed system with a reference system which delivers independently coded streams at two distinct quality levels and present detailed results on the interaction between the prediction distance, GOP size and rate-distortion performance of the resulting sequences for different head motions.

### *2.2.2 Bit-rate Allocation*

Determination of the target quality difference between the base layer MVC and the enhancement layers presents an optimization problem. The expected values of video quality experienced by the user needs to be maximized by optimally allocating the available bandwidth to the base layer MVC and enhancement layers. One dimension of this bit-rate allocation is the number of views contained in the base layer MVC representation. In Section 2.2.4 we will discuss how the receiver selects a set with $M$ views in the base layer MVC. In addition to the number of views contained in the base layer MVC, the quality difference between the base layer and enhancement layers is another parameter which determines the bit-rate allocation between the base layer MVC and the enhancement layers. The resulting bit-rate ratio between the base layer MVC and the enhancement layers is a parameter which should be used to maximize the quality of the delivered video. The Langrangian rate-distortion optimization given by

$$max\{J\}, \text{ where } J = Q + \lambda R, \tag{2.1}$$

is ideally suited for this problem. For our proposed system the bit-rate, $R$ is given by

$$R = R_{MVC} + 2 \cdot R_{ench}, \tag{2.2}$$

where $R_{MVC}$ and $R_{ench}$ are the bit-rates for the base layer MVC and an enhancement layer respectively. On the other hand, the average distortion of the received video

can be formalized as:

$$\begin{aligned} Q_{ave} = \quad & p[x=0] \cdot Q_{hi} + \\ & p[x=1] \cdot \tfrac{Q_{hi}+Q_{lo}}{2} + \\ & p[M/2-1 > x > 1] \cdot Q_{lo} + \\ & p[x=M/2] \cdot \tfrac{Q_{lo}+Q_{err}}{2} + \\ & p[x > M/2] \cdot Q_{err}, \end{aligned} \tag{2.3}$$

where $p[x=a]$ denotes the probability of a prediction error by $a$ views and $Q_{hi}$ and $Q_{lo}$ denote average PSNR quality for the enhancement layers and base layer MVC respectively, $Q_{err}$ is the quality achieved by error concealment, i.e. frame repetition for the proposed system. In Eq. (2.3), the first term denotes the quality contribution when there are no prediction errors and both eyes see the high quality views, the second term with the average of the quality of the enhancement layer and base layer MVC corresponds to a prediction error by one view when one of the eyes sees a high quality view and the other eye sees a low quality view. The third term is the quality contribution when both eyes see low quality views, which is the case when the prediction error is greater than one but still inside the $M$ views contained in the base layer MVC. The reason for the $M/2-1$ limit is due to the symmetric nature of the base layer. The fourth term denotes the case when the prediction error is exactly $M/2$, which corresponds to one eye seeing the low quality view and the view for the other eye is displayed using frame repetition from the last available frame. Finally, fifth term is the case when the prediction error is so large that both eyes see the repeated pictures from that decoded frame. This last case practically corresponds to stoppage. Since $M$ is adjusted dynamically according to the metric which will be introduced in Section 2.2.4, there is a small chance that the prediction errors will result in a complete miss. Additionally, the PSNR for frame repetition is much lower, than a high quality or a low quality view. Therefore, the contribution from the last two terms is negligible and they can be safely be dropped to simplify the closed-form

average quality formulation, which results in the following equation:

$$\begin{aligned} Q_{ave} = \; & p[x=0] \cdot Q_{hi} + \\ & p[x=1] \cdot \tfrac{Q_{hi}+Q_{lo}}{2} + \\ & p[x>1] \cdot Q_{lo}, \end{aligned} \tag{2.4}$$

which can be further simplified to:

$$\begin{aligned} Q_{ave} = \; & (p[x=0] + 0.5p[x=1]) \cdot Q_{hi} + \\ & (0.5p[x=1] + p[x>1]) \cdot Q_{lo} \\ Q_{ave} = \; & p_{hi} \cdot Q_{hi} + p_{lo} \cdot Q_{lo} \end{aligned} \tag{2.5}$$

According to Eq. (2.5) the prediction error probability distribution, $p$, is an important parameter for the resulting received multi-view video quality. The error probability distribution $p$ can be assumed to be normally distributed with zero-mean due to the Kalman-based nature of the predictor. Therefore, variance can be used to determine the probability in each term in Eq. (2.5). Since $M$ is also determined based in the prediction error variance (see. Section 2.2.4), a larger $M$ entails a larger probability of using low quality views.

Although there is no closed-form relationship between the bit-rate and average quality of a video stream, logarithmic relations of the form

$$\begin{aligned} Q_{hi} = \; & a_{hi} + b_{hi} \cdot \ln(R_{ench}) \\ Q_{lo} = \; & a_{lo} + b_{lo} \cdot \ln(R_{MVC}) \end{aligned} \tag{2.6}$$

can be employed to approximate the relationship in line with information theory fundamentals, where $a$ and $b$ parameters can be found by least squares fitting to experimental results. Thus by substituting the approximations for the quality of high quality and low quality views into Eq. 2.5, we obtain an approximation which relates bit-rates for base layer and enhancement layer to the received multi-view video quality, which can be plotted in 3-D as shown on Fig. 2.3.

If we assume a fixed available bit-rate constraint, the projection of the 3-D surface on the bit-rate constraint plane can be drawn in 2-D as shown on Fig. 2.4 and Fig. 2.5.

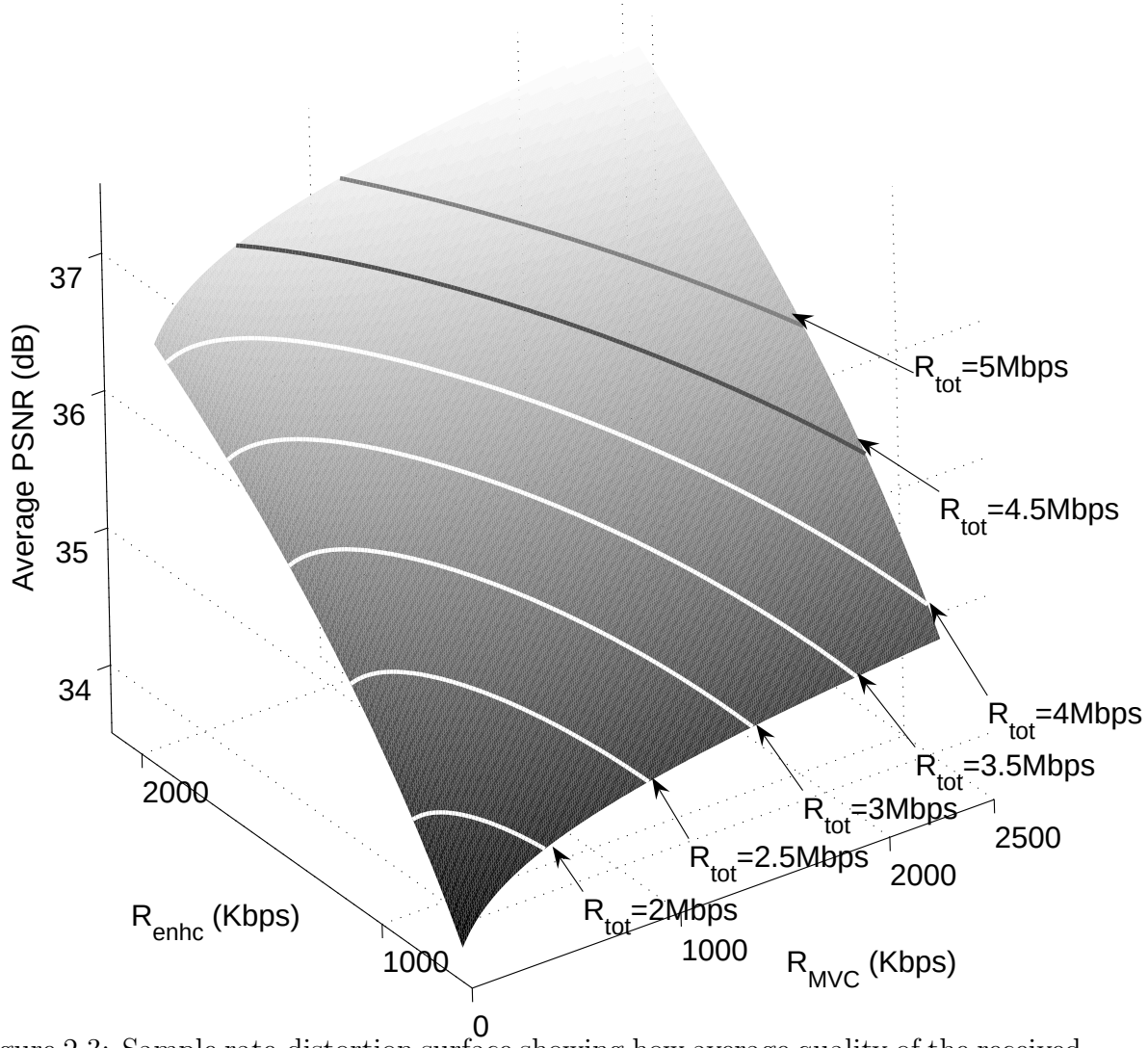



Figure 2.3: Sample rate-distortion surface showing how average quality of the received video changes with base layer MVC and enhancement layer bandwidths. The curves show total available bandwidth constraints. (75% high quality frames, 25% low quality frames)

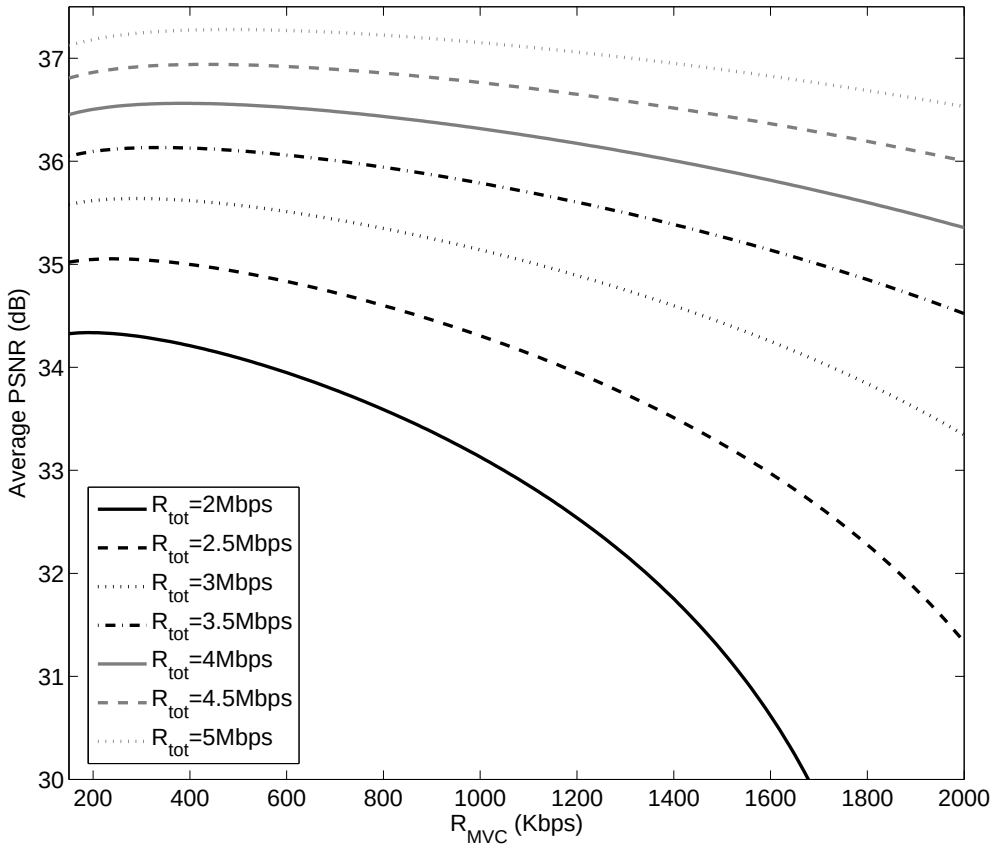



Figure 2.4: Curves showing the average received video quality vs. base layer MVC bit-rate at different available bit-rate levels and fixed prediction error probabilities. Note how the maximum quality shifts to the right with increasing total bit-rate.

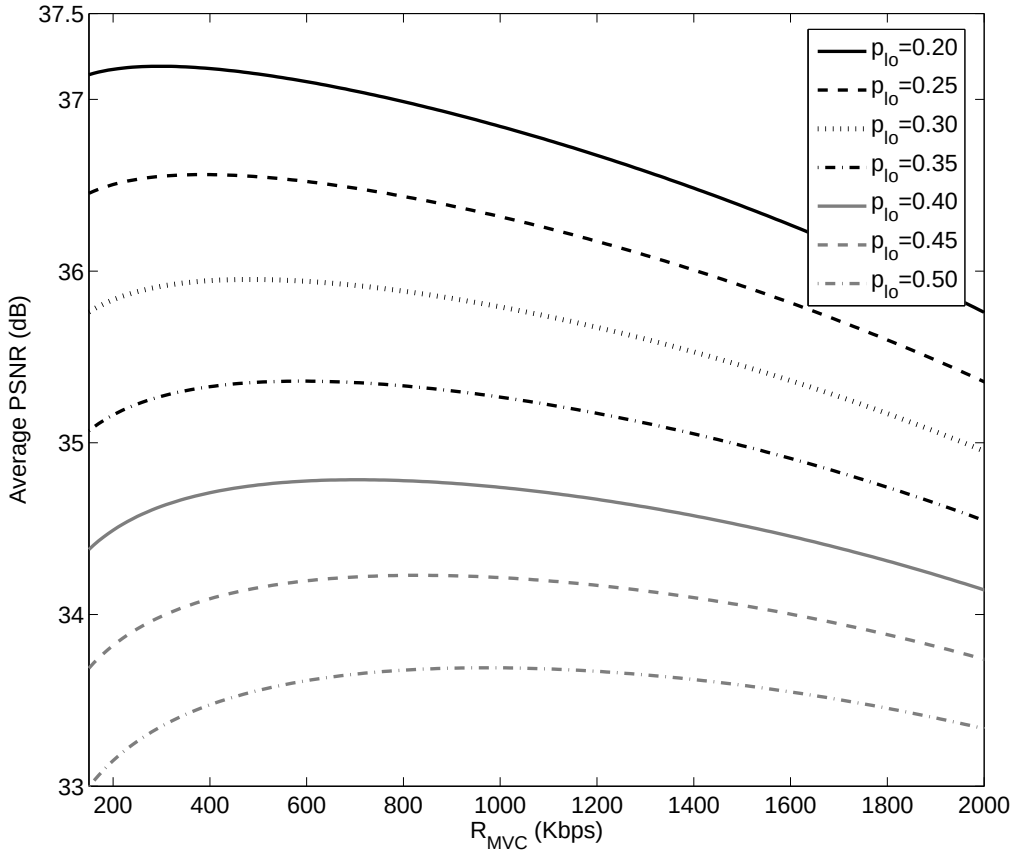



Figure 2.5: Curves showing the average received video quality vs. base layer MVC bit-rate at different prediction error distributions ($p_{lo}$ indicates probability of using a low quality view) and fixed bit-rate constraints ($R_{tot} = 4Mbps$). Note how the maximum quality shifts to the right with increasing $p_{lo}$.

### *2.2.3 Viewpoint Prediction for Prefetching*

Although a viewpoint in the world is defined by six independent parameters ($x$, $y$, $z$ for position and the Euler angles $\theta$, $\phi$, $\psi$), we only consider the situation where the movement of the user is constrained to one dimension. This assumption reflects the one dimensional physical arrangement of cameras for most multi-view sequences. Therefore the viewpoint prediction is handled by a Kalman filter with three states, in the same fashion as [13], reflecting a piecewise linear acceleration model which is shown in below.

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

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

where $T$ is the sampling interval, $x_k$ is the $k$th sample of position along the axis which reflects the physical arrangement of cameras capturing the multi-view video, and $w_k$ is the process noise, which models the acceleration change for the $k$th sampling interval, and $v_k$ is the measurement noise. We have empirically determined the measurement and process noise covariances to yield minimal error for this scenario. The employed prediction system with three states was also compared to other similar Kalman structures with one, two and four states and found to have the lowest average error variance.

At each time instance $t$, the viewer's viewpoint is sampled and a future viewpoint for the time instance $t + d$ is predicted using the Kalman predictor, where $d$ is the prediction distance. The prediction distance $d$ corresponds to the total delay between the request for a stream and the arrival of a decodable frame from the requested stream. This delay until a stream begins playing after the request, is a sum of two independent delay components: Network delay and decoding delay. The Round Trip Time (RTT) delay of the connection between the server and the client corresponds to

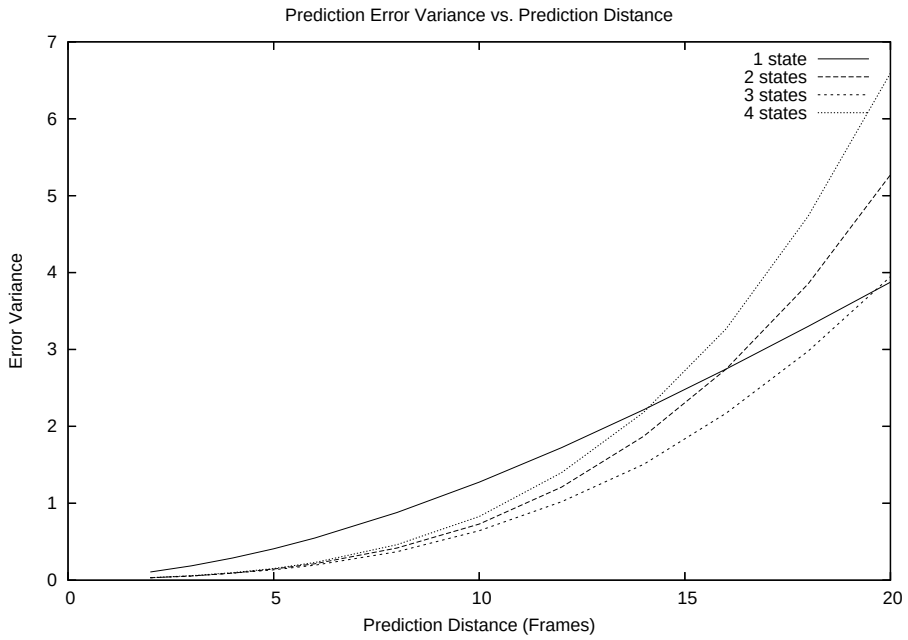



Figure 2.6: Average prediction error variance plotted against prediction distance. The Kalman predictor three states offers the lowest average error variance.

the network delay of the system, when unicast is employed. On the other hand, the join latency is the network delay in the case a multicast protocol is used to transport the streams.

The decoding delay is the wait for an independently decodable frame after the first packet of a stream arrives. This depends on the distance between independently decodable frames or the Group of Pictures (GOP) size. A larger GOP size is better for compression efficiency, but results in a longer decoding delay. However, if a frame is received, but not decodable by its play-out time, it cannot be displayed to the viewer. To accommodate for the longer decoding delays, the prediction distance must be increased as well in order to increase the probability that an independently decodable frame is received on time. Since the prediction becomes less reliable with longer prediction distances; however, a larger GOP will result in more frequent prediction errors and might cause wrong streams to be fetched from the server. In that case, the larger GOP adversely affects the user experience and deteriorate the system's perfor-

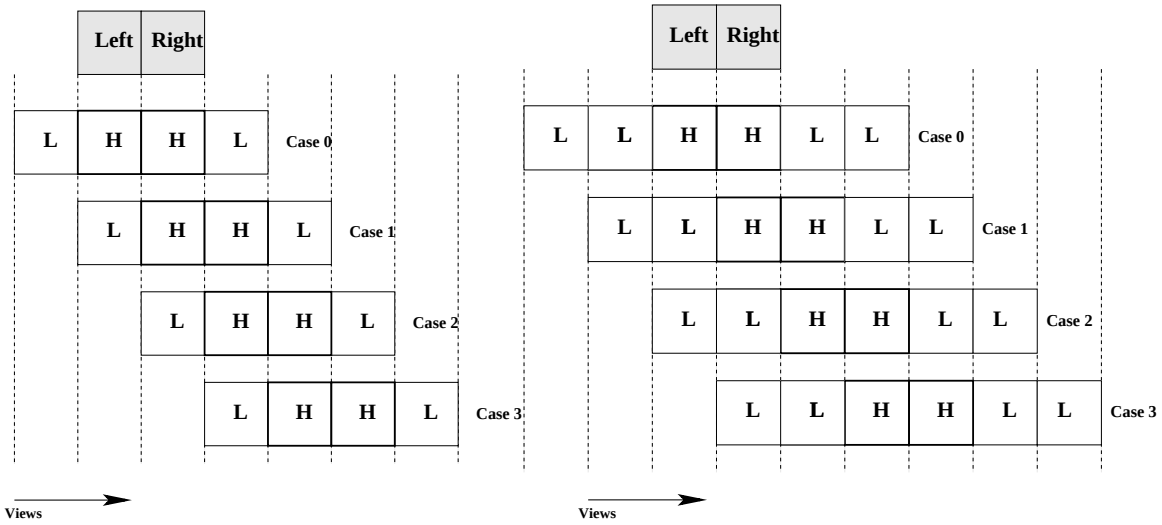



Figure 2.7: The advantage of a base layer MVC with six views (right) versus four views(left) can be seen here. The extra streams provide improved prediction error resilience.

mance. Therefore, there is a compromise between the compression efficiency provided by GOP size and the prediction errors caused by the increased system latency. The increased efficiency offered by a longer GOP can be more than offset by the errors caused due to a long prediction distance. In the presence of frequent prediction errors, a couple of redundant neighboring streams greatly improve the system performance, by providing a view to fall back on, when the prediction fails. However, they come at a bandwidth cost which offsets the coding gain provided by a larger GOP size. We present a detailed investigation of these trade-offs in Section 2.3.

### *2.2.4 Determining the Number of Views in the Base Layer*

As previously stated, a longer prediction distance results in more wrongly predicted streams because the probability of an abrupt head motion increases with the prediction distance. Thus, the performance of the prediction depends, in addition to the prediction distance, on the abruptness of the head movement as well. This means, given the same prediction distance and two viewers watching the same multi-view sequence on two different clients with the same RTT to the server, the viewer who

makes more abrupt head movements will experience more prediction errors and possibly breaks in the multi-view sequence play-out. The effects of errors can be better understood with the help of Fig. 2.7, which shows how the number of views in the base layer MVC affects performance of the system. Case 0 is when there are no prediction errors, therefore both required views are available at high quality and can be displayed to the user. Case 1 denotes the case when the prediction has failed by one view. Although the figure shows an error to the right, due to the symmetric nature of the system this also applies for the prediction errors to the left. In this case both 4-view base layer MVC and 6-view base layer MVC have one of the views in high quality and the other view in low quality. Therefore while the high quality view is displayed directly to the user, the low quality view needs to be interpolated before it can be displayed. Similarly, Case 2 denotes when the prediction error is two views, either to the left or to the right. 4-view base layer MVC does not have enough redundancy for a Case 2 error. Even though one of the views is available in low quality, the other view is not present at the client. Therefore the low quality view is interpolated and displayed, but error concealment needs to be performed for the missing view. On the other hand, 6-view base layer MVC, can gracefully handle a Case 2 error, since both required views are available at low quality. Case 3 represents an even worse situation, when the prediction has erred by three views. In that case, 4-view base layer MVC does not have any views available and the video playback will not show the requested view. On the other hand 6-view base layer MVC has one of the views in low-quality and can perform error concealment for the other view.

Therefore, it is clear that the additional views in the base layer MVC improve the performance of a selective delivery system. The additional low-quality views come at a bandwidth cost, of course, but provide a kind of insurance against prediction errors. Logically, the optimal number of low quality views, $M$, depends on the prediction distance, which in turn depends on the delay in the system, and,the abruptness of head movements. Using an abruptness metric and the delay measurements, the operating points of the system can be defined on a two-dimensional delay-abruptness plane,

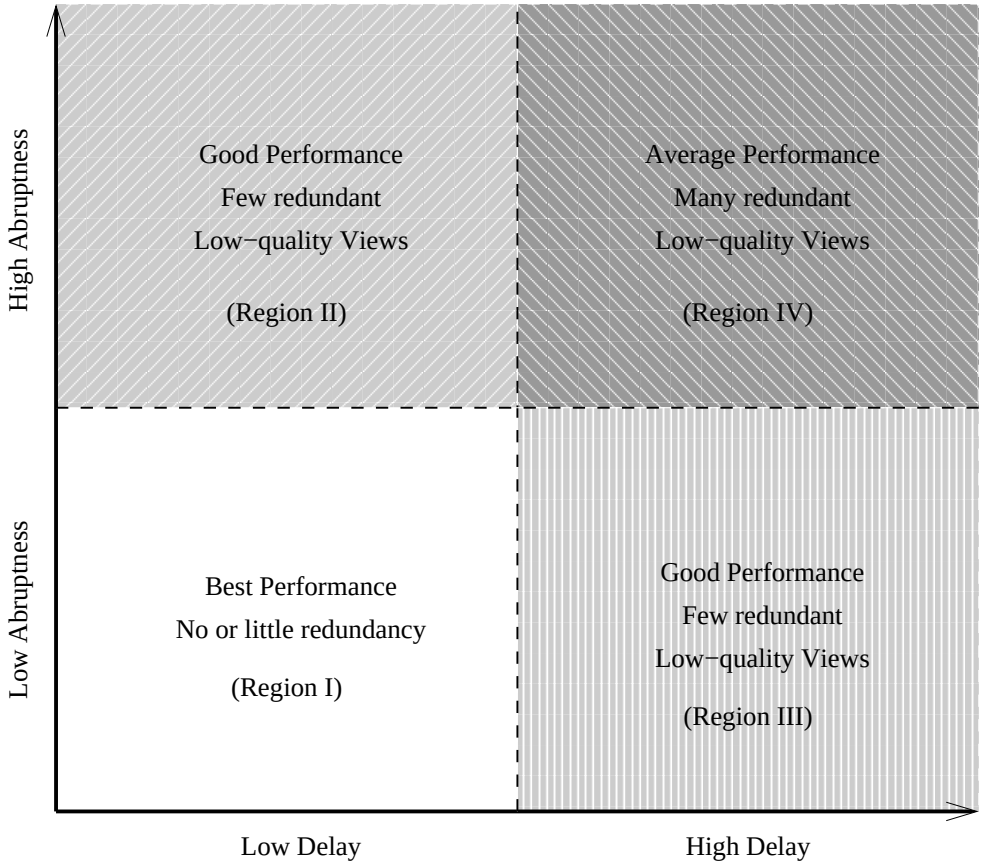



Figure 2.8: Four regions in the delay - abruptness plane.

which can be divided into four major regions as shown in Fig. 2.8 with varying suitability for the proposed selective delivery system.

If the current operating point on the delay-abruptness plane is known, the number of low-quality views can be changed to adapt to the situation. We propose to use a running average of the square of the prediction error as a metric to help determine where in the delay-abruptness plane the system is operating at. The metric for the $n$th frame can be computed as shown in Eq. (3.1):

$$R_n = \frac{1}{d} \sum_{i=n-d}^{n} (r_i - p_i)^2, \tag{2.7}$$

where $r_i$ and $p_i$ are the real and predicted positions respectively for frame $i$ and $d$ is the prediction distance. This abruptness metric corresponds to an estimate of the

error variance for the last $d$ samples, assuming the prediction error has a zero-mean distribution. This assumption is sound due to the fact the prediction follow the real values and does not introduce any drift in either direction. The rationale behind using $d$, as the window size over which the variance estimate is computed, is the fact that it corresponds to the unresponsiveness in the system, because it is a function of two delay components in the system: network delay (RTT) and decoding delay (GOP size). Therefore, our proposed metric intuitively combines two most important aspects which affect the prediction performance: head movement abruptness and the delay in fetching streams from the server and displaying them. By thresholding the computed metric, the system can estimate its operating point on the delay-abruptness plane. A low metric corresponds to Region I and a base layer MVC with a small number of views should be enough. However a larger metric would correspond to the Region II and Region III, where the base layer should contain a larger number of views. A very high metric corresponds to Region IV, where the prediction system is not able to keep up with the head movements given the total delay in the system and the proposed system might not work very well under such conditions.

## *2.3 Results*

### *2.3.1 Intended Application Scenarios*

We see two different types of 3-D display systems with different transport needs. The first candidate is projection screens which employ polarizing filters and matching glasses to show appropriate views to the eyes of the user. Although a few time-multiplexed viewers can be accommodated with the help of synchronized shutter glasses and very fast DLP projectors, these systems are best suited for single user viewing. Such single-user 3-D display systems, by definition, only need to display two views to the user. To provide a more realistic 3-D experience the user needs to be tracked and the two displayed views need to be updated accordingly. The proposed delivery system is perfectly suited for this scenario. The display systems predicts the user movements, and requests the required streams in advance. With the help of the

metric introduced in 2.2.4 the number of redundant low-quality views is determined and appropriate base layer MVC plus required enhancement streams are delivered.

The second near-future display technology makes use of lenticular lenses in order to show appropriate views to the eyes of the viewer. The lenticular lenses allow 3-D viewing without viewing aids such as polarizing glasses, and are very versatile. They can be applied to both projection systems and LCD displays. Additionally, both traditional stereo and N-view stereo is also possible using lenticular lenses. In a traditional stereo setup using an autostereoscopic display, the eyes of the user are constantly tracked and corresponding views are shown. Such a setup is analogous to a projection system. In an N-view setup, the eyes of the viewer see the appropriate two views among the spatially multiplexed N-views, as the user moves through the viewing space. Obviously, delivering only two views is not enough to drive such a display system. If multiple users are watching 3-D content on such a display, complete multi-view sequence needs to be delivered in order to display high quality content to all users. However, if only a single user is using the display, the proposed system can be employed in the following fashion. N-views are delivered with the help of base layer MVC and two enhancement streams corresponding to the user's position are delivered additionally.

### *2.3.2 A Reference System with Simulcast Streams at Two Quality Levels*

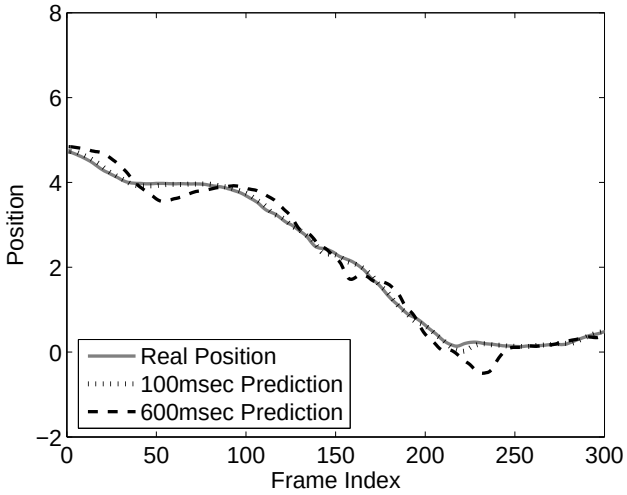
We have implemented a reference system for comparison purposes. In this scheme the server streams a multi-view video with simulcast coded views to several clients over an IP-network. Using the predicted future viewpoint, the client determines the required views and corresponding high-quality streams are requested from the server and $M$ adjacent views are streamed at a low bit-rate, where $M$ is determined using the metric defined in Section 2.2.4. If there are no prediction errors, the received streams are passed on to the display, which shows one view to each eye. However, if there was a prediction error and wrong set of high quality streams was delivered, the display system falls back to low quality streams for the views which are not delivered in high

quality. If a required view is delivered in low quality it is interpolated to original size and displayed accordingly. If the prediction error is so severe that a required view is not delivered at all, last available frame for that view is repeated as a simple means of error concealment.

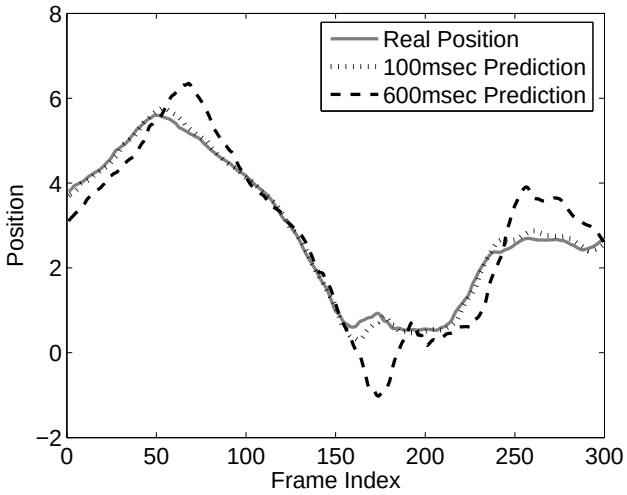
### *2.3.3 Rate-Distortion Performance*

In order to study the quality of the video delivered by the proposed system we have generated test videos using three 10sec long head motion trajectories, which were recorded using a camera based tracking system. These three trajectories can broadly be classified as slow, moderate and fast head movements. The head position prediction was performed on the recorded trajectories at various prediction distances. The slow head trajectory represents a sweeping motion from camera 5 to camera 0 over approximately 8 seconds followed by a 2 seconds long rest period. The moderate head motion represents a move from camera 4 to camera 6 in the first two seconds followed by a motion covering the whole viewing space from camera 6 to camera 0 in approximately three seconds. The viewer stays around the camera 0 from about two seconds and then quickly moves to camera 2 where he stays for the final two seconds. The fast head movement consists of a series of fast motions which cover the complete viewing space with periods of relatively low activity in between. The three trajectories and prediction performance at two different prediction distances can be seen in Fig. 2.9(a) through 2.9(c). It is clear that the prediction performance decreases as the prediction distance increases and this effect is much more pronounced when the head motion is faster.

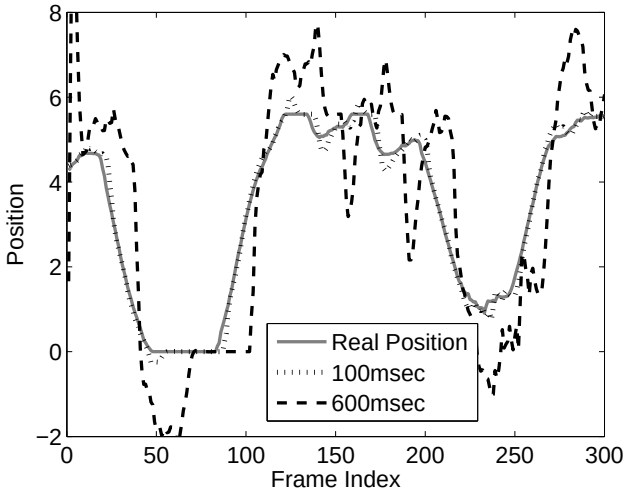
Using the predicted head trajectories, we have compared the performance of the proposed system to the reference system detailed in Section 2.3.2 and to transmitting all views at the same quality using MVC. The Race1 multi-view sequence from KDDI was used to obtain the results reported in this section. The MVC implementation from HHI was used to encode the sequence as reported in MPEG Bangkok meeting configurations. For the reference system, the views from the Race1 sequence was



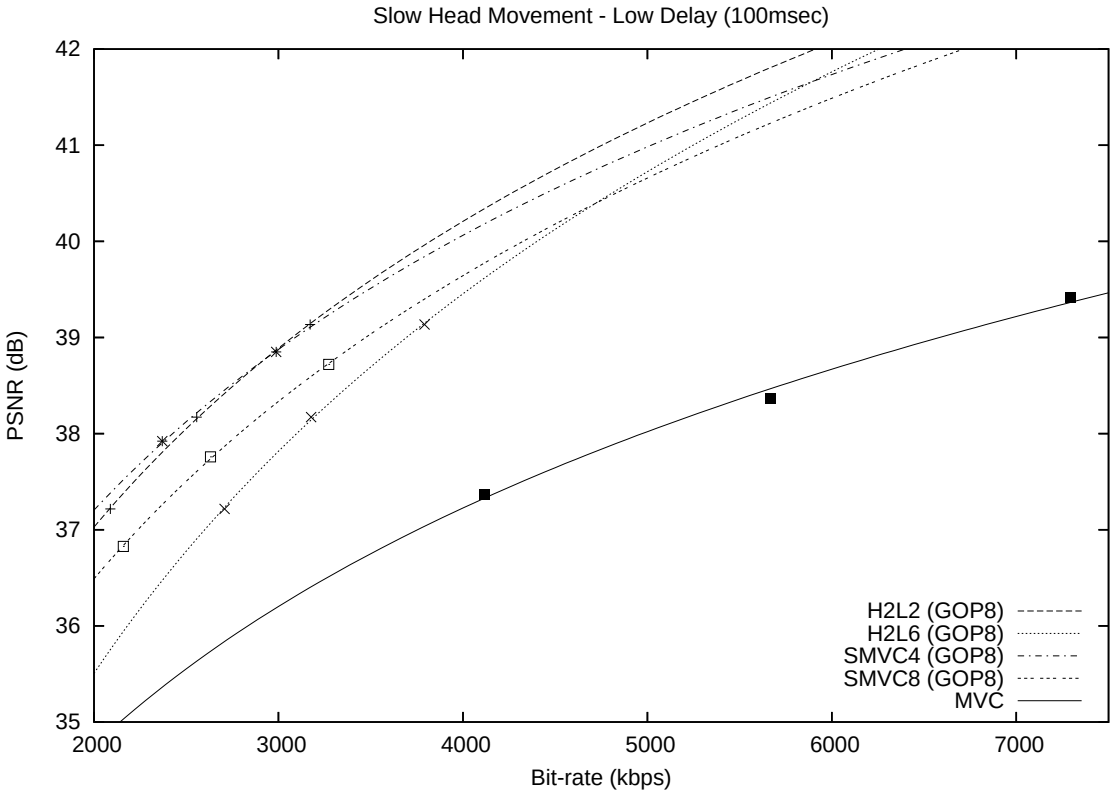

(a) slow head trajectory



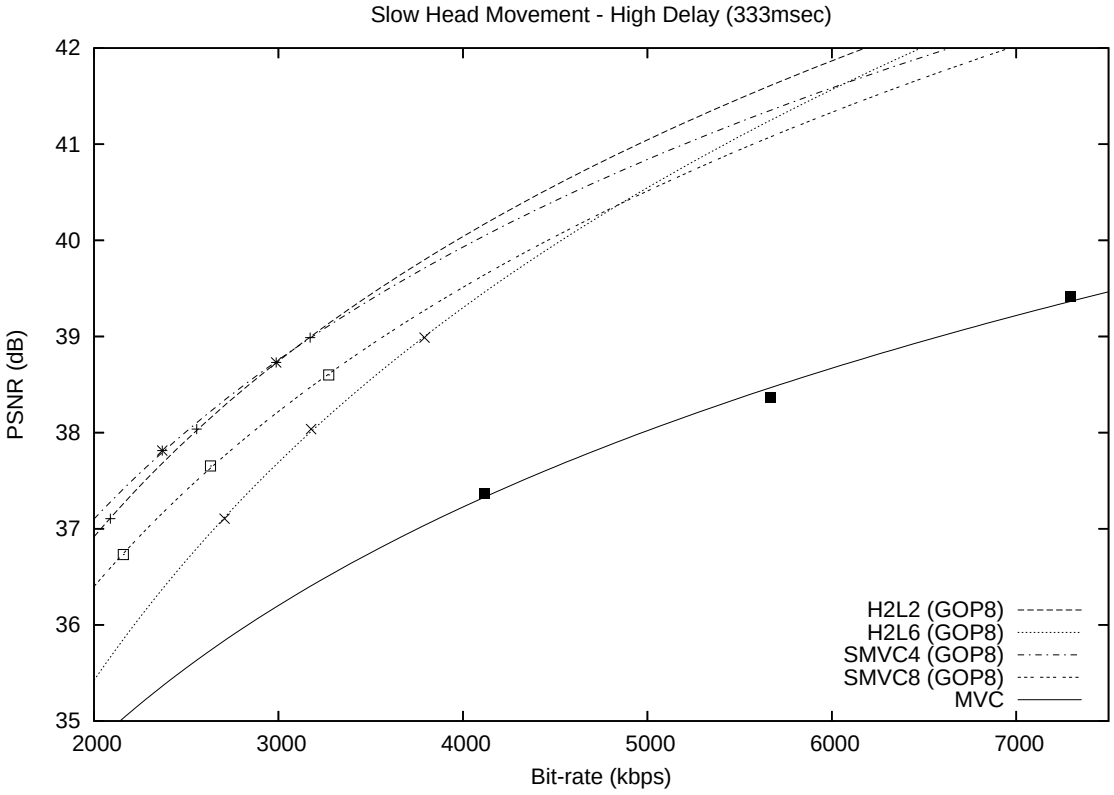

(b) moderate head trajectory



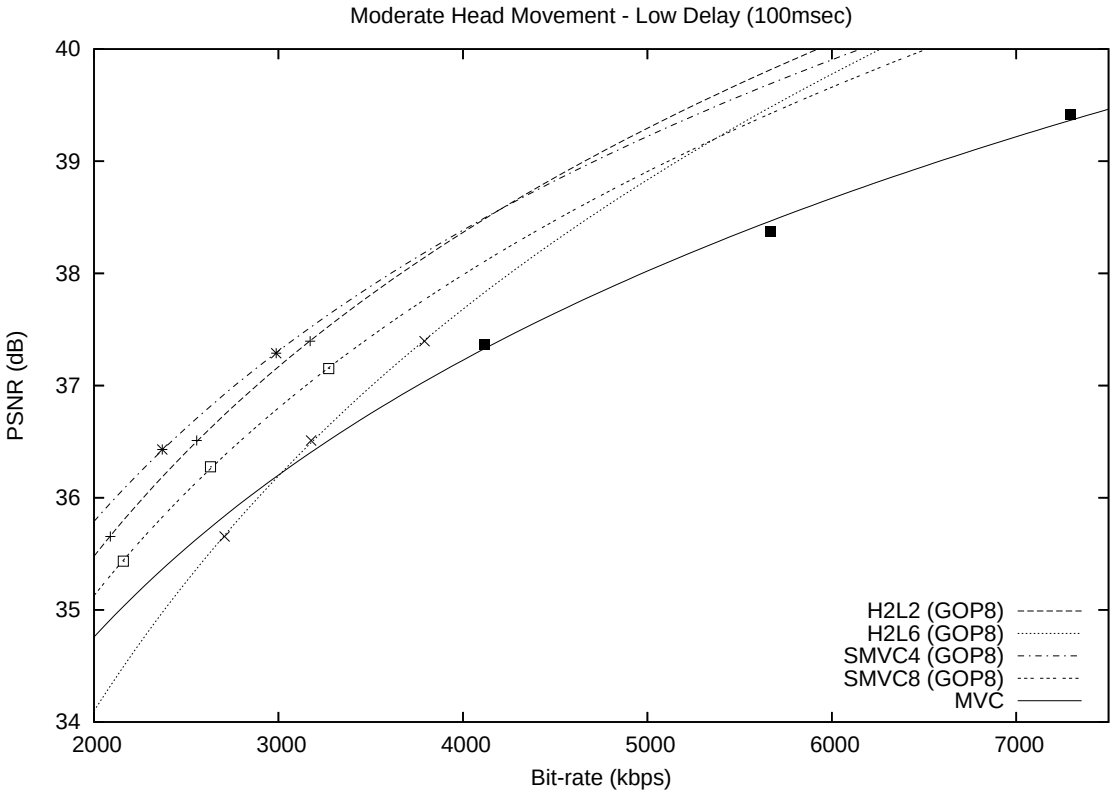

(c) fast head trajectory



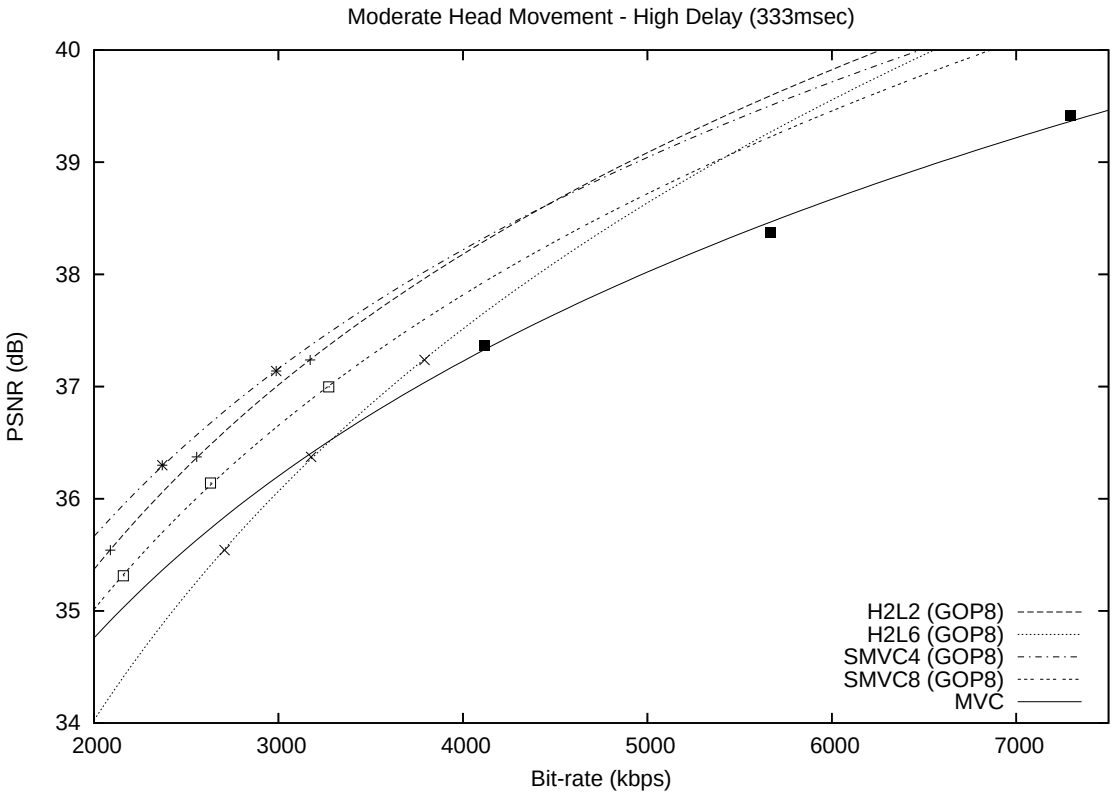

(a) low delay/slow head movement



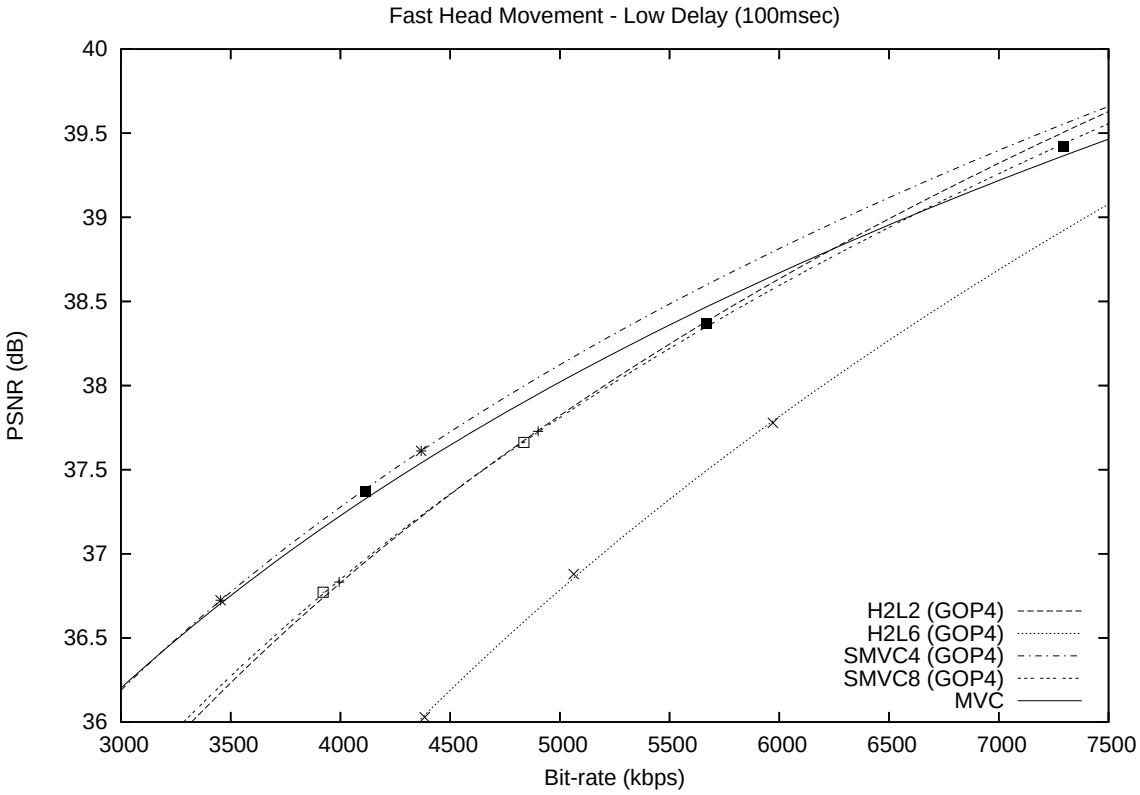

(b) high delay/slow head movement



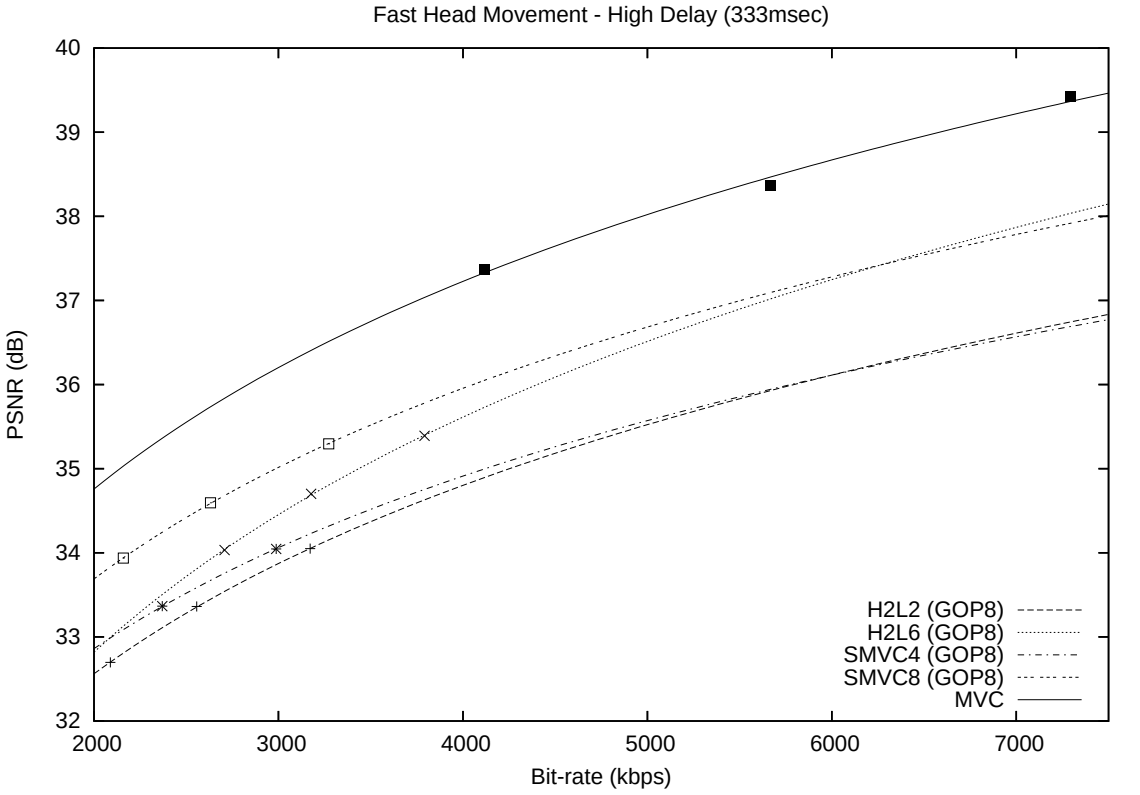

(c) low delay/moderate head movement




(d) high delay/moderate head movement




(e) low delay/fast head movement




(f) high delay/fast head movement

encoded using the reference H.264/AVC encoder at several low-quality and high-quality configurations using different QP parameters. Hierarchical B-pictures were used during encoding to optimally utilize temporal correlations. The motion search window was fixed at 96 pixels and three different GOP sizes were used: 4, 8 and 16. An IDR-picture was inserted at the beginning of each GOP, such that it becomes a stream switching point. For the proposed system the base layer MVC was encoded similar to MPEG Bangkok configurations. However the configuration files were changed to reflect the downsampled resolution and higher QP parameters for higher compression. Additionally, the JSVM SequenceFormatString parameter was modified accordingly for 4-view and 6-view base layers. The enhancement layers were encoded as described in Section 2.2.1, at various quality settings for each different base layer configuration. Similar to the reference system, the enhancement layers used three different GOP sizes: 4, 8 and 16.

In our simulations, at each time instance the client requests the corresponding stream for the predicted trajectory. After a constant RTT has passed, the requested streams begin to arrive in a decoding buffer which is as long as the GOP. If the viewpoint prediction has failed at some point between the current frame and beginning of the GOP, some of the packets needed to decode the current frame might have been not delivered, therefore, when the play-out time for a frame arrives, the buffer is checked to determine the decodability of the actually required frames in a recursive fashion. For a frame to be decodable in low quality, it must be available and all the reference frames in the base layer MVC prediction structure must be decodable as well. Additionally for a frame to be decodable in high quality, it needs to be available in the enhancement stream, decodable in low quality and its reference frames in the enhancement stream must be decodable as well. During test video generation the system first checks if the required frame at a particular time instance is decodable in high quality, if not it falls back to low quality, in the case the frame is not decodable in low quality as well, last displayed frame is repeated as a crude error concealment method. Another possible error concealment method is displaying the frames for

the original view if the frames for the requested view are not available, practically ignoring a requested view change. Frame repetition causes video freezing during error periods but reflects user position more accurately, on the other hand displaying what is available does not have to deal with the problem of freezing but is less responsive in the case of prediction errors. The subjective evaluation of these error concealment methods is outside the scope of this chapter. MVC test sequences are generated in a similar fashion but they are not affected by missed streams since all views are available at each time instance. The PSNR values for the final recorded sequences are computed with respect to reference sequences, which are generated using the same viewpoint data assuming no delays, perfect prediction, perfect decodability and no encoding distortion, i.e. for the PSNR reference sequences every frame in every view is always available with no encoding distortion and perfect head position information. Figures 2.10(a) through 2.10(f) demonstrate the Rate-Distrotion characteristics of the proposed system when compared to the reference system and to standard MVC. As it can be seen from these figures the performance of selective delivery systems, both the reference system and the proposed system, deteriorate with increasingly faster head movements and longer delays. Additionally, the proposed system outperforms the reference system at lower bit-rates but usually the reference system is slightly better at higher bit-rates. However, as the operating conditions get worse, the proposed system performs increasingly better when compared to the reference system, but the advantage in relation to MVC starts to decrease. MVC outperforms both selective delivery schemes for fast head movements with high delay.

*Compression efficiency, latency trade-off*

As mentioned in Section 2.2.3, there is a trade-off between the compression efficiency offered by a longer GOP and introduced latency due to the decoding delay. We have experimentally found that a GOP of 16 frames results in a poorer received video rate-distortion performance. The poor performance of 16 frames GOP is caused by two factors: the prediction distance can be increased to compensate for the GOP

size, which results in more frequent prediction errors, or a short prediction distance can be used but then some of the frames might be missed if the beginning of the GOP is missing. We have found a GOP of 8 frames to be best suited compromise for the proposed selective delivery system. Although a 4-frame GOP offers better latency performance, it is not enough to offset the compression efficiency lost to frequent IDR-frames, except for very fast head movements where quick stream switching is very important (see. Fig. 2.10(e)). However it should be noted that in such situations the proposed selective delivery system offers little, if at all, performance gain over sending the whole MVC stream in high quality. Therefore a GOP of 8 frames is optimal whenever the proposed system is feasible.

### *2.3.4 Optimal Mode Selection*

In light of the results presented in Section 3.3.3, the received video quality clearly depends on the abruptness of the head motion and the delay in the system supporting our reasoning in Section 2.2.4. In order to address the optimization problem defined in Section 2.2.2, we have computed the prediction error variance metric as proposed in Section 2.2.4 for the predicted head trajectories. Fig. 2.11 shows how the metric changes with prediction distance for different head trajectories. It can be clearly seen that the prediction becomes very unreliable for faster head trajectories. Fig. 2.12 is a scatter plot of all available multi-view video delivery experiments. It shows three sets of data points for four, six and eight views in the base layer. In order to obtain these data points a filtering operation was performed on the data points to remove samples which contain more views than strictly necessary, i.e. a data point was removed if and only if it provides exactly the same average quality with a base layer with fewer views at the same encoding settings. This filtering method ensures that the base layer with fewer views, and thus lower bit-rate, is favored at the quality level. These removed data points correspond to experiments where additional views in the base layer lead to no PSNR improvement at all, when all other conditions are the same. The decision lines were fitted to the data points in the least squares sense and denote the average

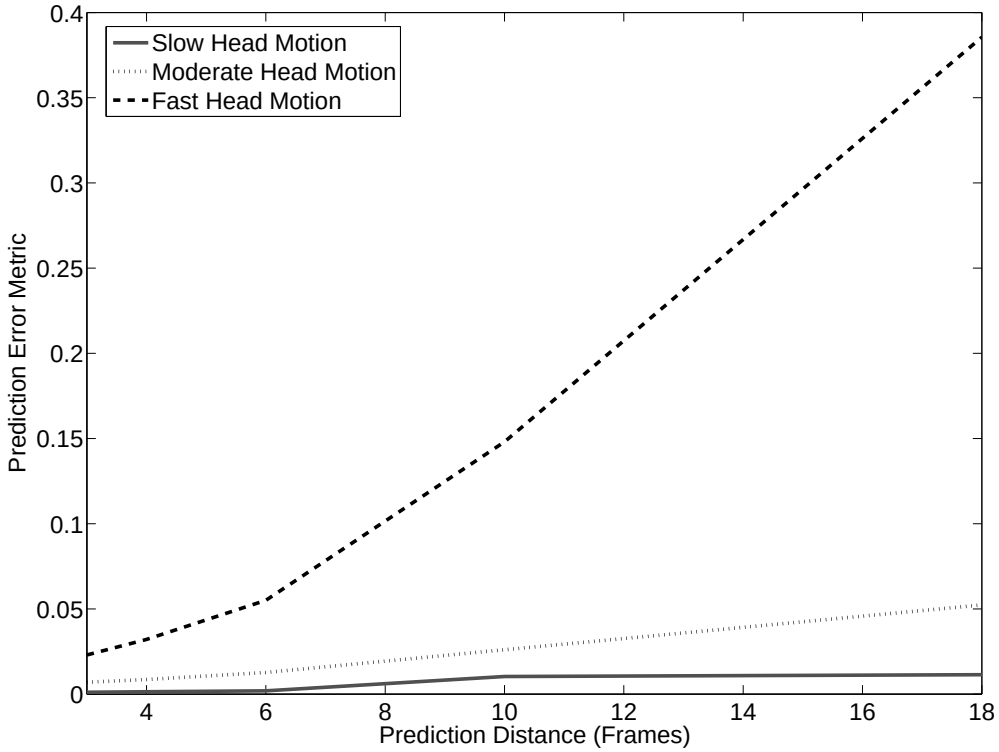



Figure 2.11: Prediction error metric vs. the prediction distance for different head trajectories.

expected quality for a prediction error metric value. Therefore, for a prediction error metric value, the highest line is the optimum line. As it can be seen from the figure, 0.15 is found to be the decision boundary between SMVC4 and SMVC6. Four views in the base layer are enough, if the prediction error metric is lower than 0.15. Six views are better otherwise. There is no intersection between the decision line of SMVC6 and SMVC8, which suggests that the bandwidth cost of two additional views is not feasible, since the prediction hardly ever fails badly enough that the required views are not contained in a base layer with six views.

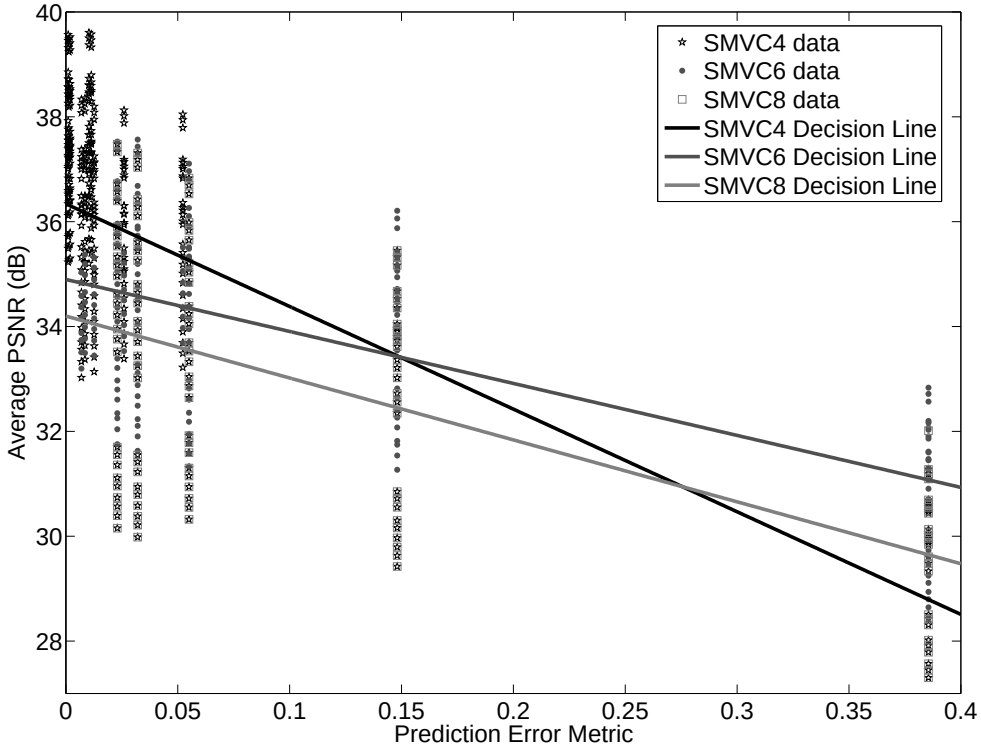



Figure 2.12: Average quality vs. prediction error metric. Data points denote that a base layer with fewer views is not worse for that particular prediction error metric value at the same quality and GOP settings. The highest decision line is selected for a given prediction error metric.

## *2.4 Conclusions*

We presented an encoding scheme for multi-view videos, which makes use of MVC and SVC concepts. Additionally, we have introduced a novel transmission method for the proposed encoding scheme. The proposed transmission scheme features selective delivery of views, such that only the views are delivered which are required to display the user's current view. The proposed scheme also contains methods to predict the user's future head positions and to adaptively control the number of low quality views in the base layer according to the prediction error variance. We have shown that the proposed system outperforms MVC in the sense of transmitted bits for most operating conditions and is up to 3dB more efficient in some cases. Total size of the compressed representation, however, is larger. It was found out that the low quality neighboring streams are well worth their bandwidth cost, since they allow play-out of the stereo video to continue when the viewpoint prediction errs by one view. We have also observed that accurate viewpoint prediction is very important for the performance of the system.

# Chapter 3

# DELIVERY OF MULTI-VIEW VIDEO USING APPLICATION-LAYER MULTICAST

## *3.1 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, less far-fetched varieties of interactive 3D entertainment can be in living rooms in the very near future.

The first products to offer interactive 3D will most probably based on a technology to show two 2-D views such that left and right eyes of the viewer see the scene 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.

### *3.1.1 3D Representations*

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 3-D scene is highly dependent on the scene geometry and is usually very high for real-time applications [6]. Moreover accurately capturing 3-D 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 viewpoints. The idea behind the IBR systems is the seven dimensional plenoptic function, which describes all potentially available optical information in a given region [16]. Various IBR systems, such as light fields [6], are simplifications of this plenoptic function. Pure IBR systems do not assume an explicit 3-D model of the scene. However as suggested by Kang et al. in [17], 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.

The reconstruction quality of IBR systems depends on the sampling density in the camera plane or the availability of the scene geometry. As a result IBR representations without some form of geometry information require large number of cameras to capture the scene and generate enormous amounts of data for a good reconstruction quality. Even though previous research [18] [19] [20] suggests very high compression rates, these come at the cost of difficulty in interactive viewing. It appears unlikely to expect compression algorithms to be developed, which allow interactive viewing and reduce the data rate to levels that make it feasible to stream over domestic broadband connections in the near future.

Other IBR systems, such as multi-view videos, make use of the geometry information to reduce the required amount of data. The most common form of geometry information used in such systems is depth or disparity maps. Previous work [2] has shown 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 of the original bit rate of a single-view video. Therefore, to send such high-bandwidth multi-view video over existing broadband connections, there is a need for novel networking schemes,

which make efficient use of the available bandwidth.

### *3.1.2 Multicasting Solutions*

Multicasting is a method to efficiently transport packets from one or more senders to a group of interested clients. Multicasting paradigm aims to prevent sending of duplicate packets to clients in the network in order to utilize the network resources more efficiently. In the case of network-layer multicast, the sender sends every packet only once. These packets get duplicated at multicast-enabled routers as needed and forwarded to other members of the multicast group. Even though network-layer multicasting is almost always the most efficient method to distribute information to a set of interested clients, it is not widely deployed in practice because it requires multicast capable routers. Furthermore, although most of the new routers are multicast capable; security and other operational concerns discourage network operators from enabling the multicast functionality. Therefore, large parts of the Internet are cut off from network-layer multicast capability [21].

The alternative to duplicating and forwarding packets at routers is to shift that functionality to 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 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 traveling through more hops than network-layer multicast, and some physical links carry some duplicate packets [22]. Two other important factors which need to be considered when assessing the merits of an application layer multicast protocol are join-latency and the amount of control traffic overhead to construct and maintain the distribution tree.

Various application layer multicast protocols have been reported to be applicable to media transport. End System Multicast (ESM) uses NARADA protocol [22] to deliver video streams to viewers around the globe. A loss resilient variant of NICE protocol with probabilistic loss recovery was studied for video delivery over RTP in [23].

In [24], Setton et al. present results for a video streaming system using application layer multicast, which introduces a rate-distortion optimized approach to up-stream bandwidth allocation at peers. Moreover Hosseini and Georganas have demonstrated a 3-D video conferencing system [25] utilizing an application layer multicast protocol, and introduced the awareness-driven video concept to overcome the up-stream bandwidth limitations for multi-user video conferencing applications by only delivering videos for some of the users, which is essentially parallel to the selective delivery idea, which is further elaborated later in this chapter.

Our motivation in this chapter is to reduce the bandwidth requirements of multi-view video delivery by utilizing application layer multicast. In [26], McCanne et al show that 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. This concept can readily be adapted to the transport of multi-view video streams, where each view in the multi-view video can be assigned to a different layer and the viewer can choose which views they need to receive and join the corresponding multicast group. As a preliminary proof-of-concept, we proposed network-layer multicast as a possible solution for the transport of dynamic light fields, where each view is streamed to a different IP-multicast address [9]. In that preliminary work, a viewer's client joins the appropriate multicast groups to only receive the 3-D information relevant to its current viewpoint.in this chapter we build on that approach and present a system which employs NICE application layer multicast protocol to reduce the bandwidth requirements, and therefore costs, of broadcasting multi-view content over a network.

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

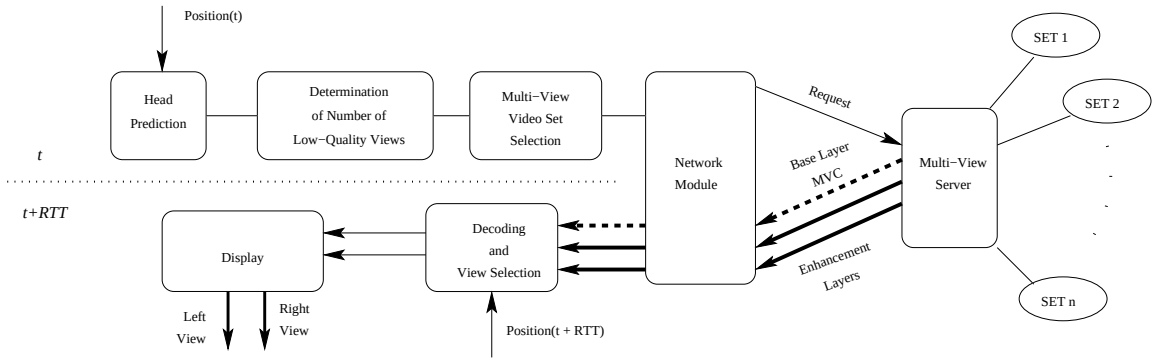



Figure 3.1: The 3DTV multicast streaming system overview

## *3.2 System Description*

We propose to deliver views of a multi-view sequence over NICE application layer multicast protocol at two quality levels using a scalable encoding: $M$ views out of the total $N$ of the multi-view sequence are encoded using MVC at low bit-rate as a base layer. On top of this base layer, enhancement layers are encoded. The system first determines the user's head position and a Kalman-filter based prediction subsystem predicts the user's head position $d$ frames into the future. Using the predicted positions, an error metric is computed at the client to determine the number of views in the base layer, $M$. The bandwidth available to the user is assumed to be limited, therefore as $M$ increases an increased proportion of the fixed bandwidth is allocated to the base layer. This necessitates an intelligent allocation of bandwidth between the base layer and the enhancement layers. The server has several sets of the same multi-view video. Each set is encoded using a different $M$-number and has different bit rate allocation ratios between the base layer and the enhancement layers. The client switches to the appropriate set of streams according to the user's predicted head position, its bandwidth and current $M$-number, which in turn is determined using a metric based on an estimate of prediction error variance.

In this section, the server side issues are first described in Sections 3.2.1 and 3.2.2. Then the details of the multicast architecture are presented in Section 3.2.3 and finally

client-side features are described in the Sections 3.2.4 and 3.2.5

### *3.2.1 MVC Base Layer with Simulcast Enhancement Layers*

The proposed approach to selectively transport the required views over application layer multicast is using MVC coding with additional enhancement layers exactly, as detailed in Section 2.2.1. This involves encoding all views together using MVC as a lower bit-rate base layer. The base layer is spatially downsampled to reduce the encoded video size. In addition to this base layer, an enhancement layer is generated for each view as follows: first the decoded video for each view in the MVC stream is upsampled to the full resolution of the original video and then the difference between the original videos and the decoded MVC videos are encoded using AVC/H.264. Alternatively the spatial scalability features of the emerging SVC standard can be used for this purpose in a similar fashion. These enhancement layers are independently coded to provide greater flexibility in switching from one view to the next. The coding structure of the proposed MVC plus enhancement layers can be seen in Fig. 2.2. This proposed structure benefits from the coding gain offered by MVC, while providing significantly greater flexibility in view selection, such that a user receiving the base layer and the enhancement layer for view $n$, is able to see it at a high quality, while all other views would be available at a lower quality.

### *3.2.2 Bit-rate Allocation*

Determination of the target quality difference between the base layer MVC and the enhancement layers presents an optimization problem. The expected values of video quality experienced by the user needs to be maximized by optimally allocating the available bandwidth to the base layer MVC and enhancement layers. One dimension of this bit-rate allocation is the number of views contained in the base layer MVC representation. In Section 2.2.4 we will discuss how the client selects a set with $M$ views in the base layer MVC. In addition to the number of views contained in the base layer MVC, the quality difference between the base layer and enhancement layers

is another parameter which determines the bit-rate allocation between the base layer MVC and the enhancement layers. The resulting bit-rate ratio between the base layer MVC and the enhancement layers is a parameter which should be used to maximized the quality of the delivered video.

Assuming that the prediction error is normally distributed with zero mean, the variance can be used to determine the probability of both eyes seeing two low quality views, which corresponds to low perceived stereo video quality. Since $M$ is also determined based on the prediction error variance, a larger $M$ entails a larger probability of using low quality views. Therefore, to obtain the best possible average quality, it is best to increase the relative quality of the base layer MVC when it contains more views. It should also be noted that as $M$ increases the bandwidth allocated for the base layer MVC increases not only due to the increased number of views contained in the stream but also because of higher quality encoding.

### *3.2.3 Application Layer Multicast Distribution Network*

The multi-view video sets are streamed to the viewers using an application layer multicast distribution network based on the NICE protocol. Although it theoretically possible to dynamically construct this distribution network between the end-users, due to the high band-width requirements of the multi-view video, it is unrealistic to expect viewers to allocate their precious up-stream bandwidth to forward multi-view video data to other peers. Additionally, unlike dedicated servers or network equipment, the end-users in cooperative systems are inherently less reliable. Random and ungraceful disappearance of the end-user nodes poses a problem for the performance of the delivery tree. Therefore, using a hierarchical approach is better suited for this problem. As shown in Fig. 3.2, the multi-view video (MVV) server sends the streams to several "professional" multicast peers over the distribution network. These peers forward data either to other peers over the distribution network or to connected end-users over unicast. The peers also forward and process the requests for streams sent out by the end-user clients.

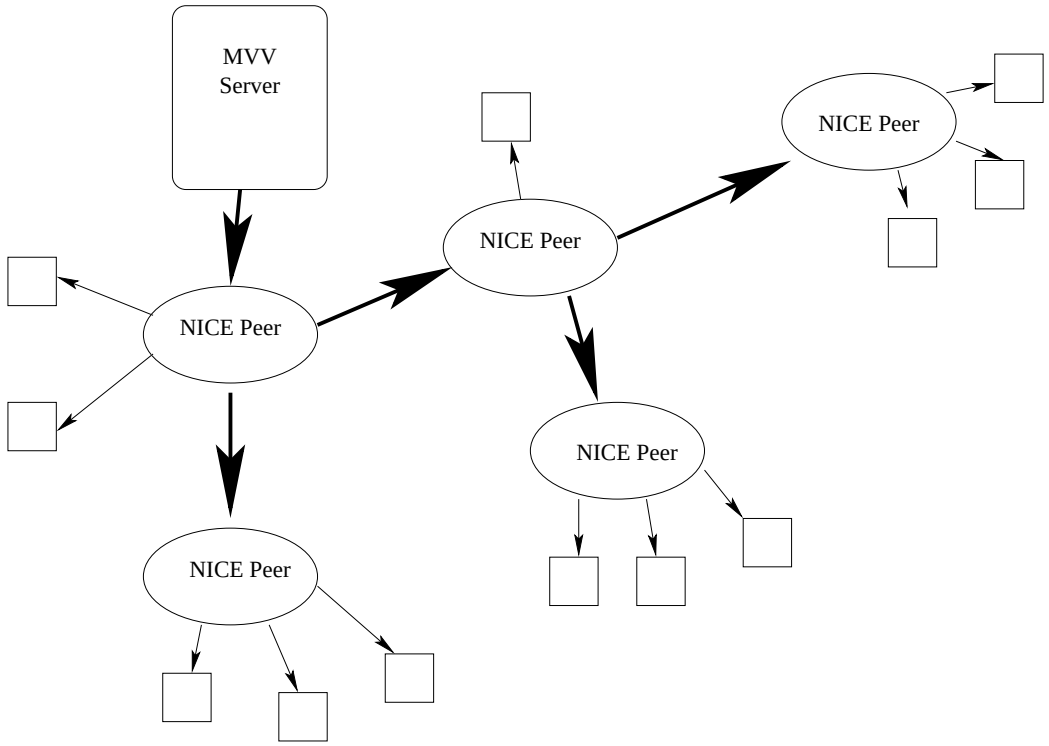



Figure 3.2: A sample distribution tree for a small network. The ovals represent the NICE application layer multicast peers and the small squares represent the end-users. The bold arrows denote the multi-view data on the multicast distribution network and the small arrows denote packets delivered from NICE peers to end-users over unicast.

In NICE protocol, members of a multicast group organize themselves into small clusters using ping roundtrip time measurements. These clusters also form the lowest layer, $L_0$, in the 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 cluster leaders of clusters in $L_n$ become members of $L_{n+1}$ until one 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.

An independent NICE hierarchy is maintained for each base layer MVC and enhancement stream in the multi-view video distribution. The end-user clients send out requests for these streams according to their current viewpoint and head prediction error variance as described in Section 2.2.2. Each end-user client has a list of geographically close application layer multicast peers. When a particular stream is required, the end-user client contacts one of the known peers, which is estimated to have the best possible connection to the client and requests the stream. If the peer in question is already a member of the corresponding distribution tree, it already has the requested stream and starts forwarding packets to the new client immediately. If, however, that peer is not a member of that multicast group, it contacts the bootstrap node and requests to join the corresponding 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 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.

The 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 join-latency of the NICE protocol is fairly low. In addition to the dynamic join-latencies, the time of flight changes dynamically for each packet. This dynamic nature of time of flight is caused by both the physical router queues in the network and the possible path changes in the overlay network. Therefore the proposed selective delivery system

needs to adapt to the dynamic and random nature of the application layer multicast delivery network. We propose to handle the variations in delay by a head position prediction system, which is the topic of the next subsection.

### *3.2.4 Viewpoint Prediction for Prefetching*

Although a viewpoint in the world is defined by six independent parameters ($x$, $y$, $z$ for position and the Euler angles $\theta$, $\phi$, $\psi$), we only consider the situation where the movement of the user is constrained to one dimension. This assumption reflects the one dimensional physical arrangement of cameras for most multi-view sequences. Therefore the viewpoint prediction is handled by a Kalman filter with three states, in the same fashion as [13].

At each time instance $t$, the viewer's viewpoint is sampled and a future viewpoint for the time instance $t + d$ is predicted using the Kalman predictor, where $d$ is the prediction distance. The prediction distance $d$ corresponds to the total delay between the request for a stream and the arrival of a decodable frame from the requested stream. This delay until a stream begins playing after the request, is a sum of two independent delay components: Network delay and decoding delay. The Round Trip Time (RTT) delay of the connection between the server and the client corresponds to the network delay of the system, when unicast is employed. On the other hand, the join latency is the network delay in the case a multicast protocol is used to transport the streams.

The decoding delay is the wait for an independently decodable frame after the first packet of a stream arrives. This depends on the distance between independently decodable frames or the Group of Pictures (GOP) size. A larger GOP size is better for compression efficiency, but results in a longer decoding delay. However, if a frame is received, but not decodable by its play-out time, it cannot be displayed to the viewer. To accommodate for the longer decoding delays, the prediction distance must be increased as well in order to increase the probability that an independently decodable frame is received on time. Since the prediction becomes less reliable with longer

prediction distances; however, a larger GOP will result in more frequent prediction errors and might cause wrong streams to be fetched from the server. In that case, the larger GOP adversely affects the user experience and deteriorate the system's performance. Therefore, there is a compromise between the compression efficiency provided by GOP size and the prediction errors caused by the imposed delay. The increased efficiency offered by a longer GOP can be more than offset by the errors caused due to a long prediction distance. In the presence of frequent prediction errors, a couple of redundant neighboring streams greatly improve the system performance, by providing a view to fall back on, when the prediction fails. However, they come at a bandwidth cost which offsets the coding gain provided by a larger GOP size. We present a detailed investigation of these trade-offs in Section 2.3.

### *3.2.5 Determining the Number of Views in the Base Layer*

As previously stated, a longer prediction distance results in more wrongly predicted streams because the probability of an abrupt head motion increases with the prediction distance. Thus, the performance of the prediction depends, in addition to the prediction distance, on the abruptness of the head movement as well. This means, given the same prediction distance and two viewers watching the same multi-view sequence on two different clients with the same delay to the server, the viewer who makes more abrupt head movements will experience larger prediction errors and possibly breaks in the multi-view video play-out.

Therefore, it is clear that the additional views in the base layer MVC improve the performance of a selective delivery system in the presence of large prediction errors. The additional low-quality views come at a bandwidth cost, of course, but provide a kind of insurance against prediction errors. Logically, the optimal number of low quality views, $M$, depends on the prediction distance, which in turn depends on the delay in the system, and,the abruptness of head movements. Using an abruptness metric and the delay measurements, the operating points of the system can be defined on a two-dimensional delay-abruptness plane, which can be divided into four major

regions as shown in Fig. 2.8 with varying suitability for the proposed selective delivery system.

If the current operating point on the delay-abruptness plane is known, the number of low-quality views can be changed to adapt to the situation. We propose to use a running average of the square of the prediction error as a metric to help determine where in the delay-abruptness plane the system is operating at. The metric for the $n$th frame can be computed as shown in Eq. (3.1):

$$R_n = \frac{1}{d} \sum_{i=n-d}^{n} (r_i - p_i)^2, \tag{3.1}$$

where $r_i$ and $p_i$ are the real and predicted positions respectively for frame $i$ and $d$ is the prediction distance. This abruptness metric corresponds to an estimate of the error variance for the last $d$ samples, assuming the prediction error has a zero-mean distribution. The rationale behind using $d$, as the window size over which the variance estimate is computed, is the fact that it corresponds to the unresponsiveness in the system, because it is a function of two delay components in the system: network delay (RTT) and decoding delay (GOP size). Therefore, our proposed metric intuitively combines two most important aspects which affect the prediction performance: head movement abruptness and the delay in fetching streams from the server and displaying them. By thresholding the computed metric, the system can estimate its operating point on the delay-abruptness plane. A low metric corresponds to Region I and a base layer MVC with a small number of views should be enough. However a larger metric would correspond to the Region II and Region III, where the base layer should contain a larger number of views. A very high metric corresponds to Region IV, where the prediction system is not able to keep up with the head movements given the total delay in the system and the proposed system might not work very well under such conditions.

## *3.3 Results*

### *3.3.1 Intended Application Scenarios*

In addition to the intended application scenarios listed in Section 2.3.1, the application layer multicast delivery system detailed in this chapter is intended to reduce the bandwidth requirements at the server, while providing comparable rate-distortion performance to the unicast system described in Chapter 2.

The proposed system fits perfectly into the concept of displaying multi-view video on autostereoscopic displays or projection screens. Selective delivery with adaptive number of views in the base layer MVC aims to keep the bandwidth requirements low at the clients, whereas application-layer multicast with professional peers distributes the bandwidth requirements and reduces the costs of delivering 3-D content to numerous viewers.

### *3.3.2 Network Experiments*

The NICE protocol has been modified such that hosts can simultaneously be members of multiple parallel multicast hierarchies. Network infrastructure of the ISP firm Tiscali in Germany as shown in Fig. 3.3, was recreated in the Emulab network testbed [27] using measurements obtained from the rocketfuel topology database [28]. On that emulated topology we let end-users connect to the multi-cast peers in a random fashion in order to simulate worst case conditions. This random join and leave period was simulated for half an hour at approximately 45 total multicast membership changes per minute. During this period NICE activity was logged and join-latency, time-of-flight and packet loss information was later compiled from the logs.

As previously mentioned, application layer multicast has a dynamic join-latency and time-of-flight for packets. As the histograms in Fig. 3.4(a) and Fig. 3.4(b) suggest these random processes seem to be governed by multi-modal normal distributions. Therefore, we have modeled these delay components using two separate probability distributions obtained by fitting Gaussian Mixture Models (GMM) to the experimen-

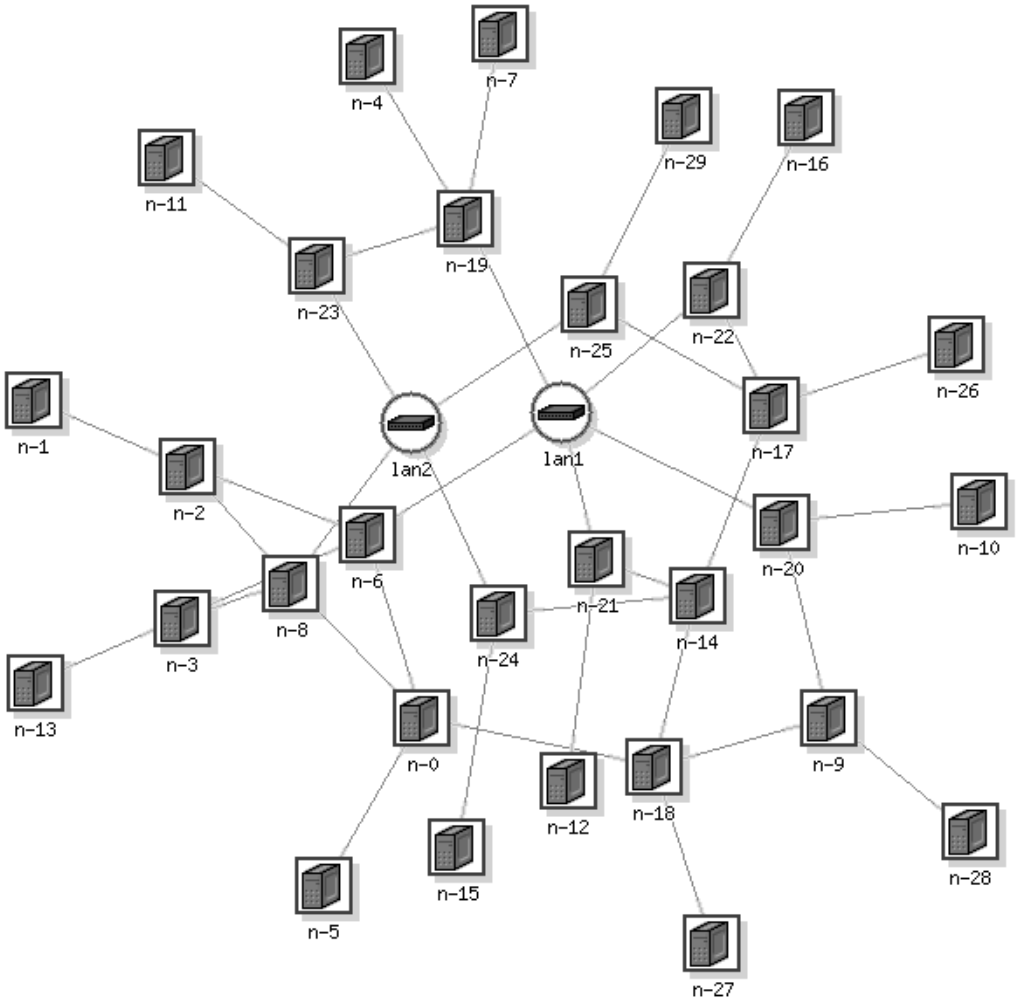
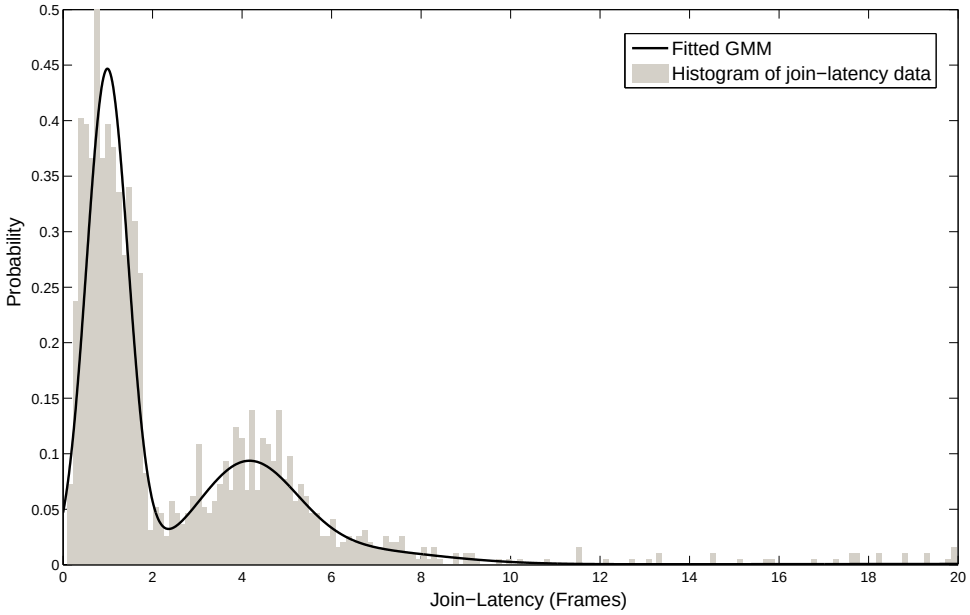
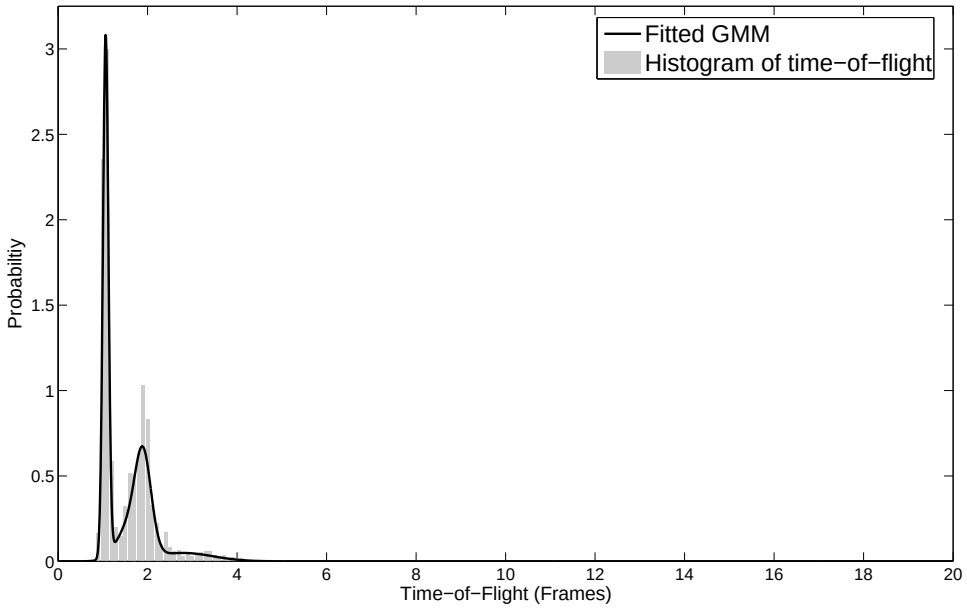



Figure 3.3: The backbone topology of ISP firm Tiscali in Germany. (Screen capture from Emulab user interface)




(a) join latency




(b) time of flight

Figure 3.4: The histogram of measured join-latency [3.4(a)]and time-of-flight [3.4(b)] delays together with the probability distribution fitted using GMM

tal join-latency and time-of-flight measurements. The fitted distributions can been seen in Fig. 3.4(a) and Fig. 3.4(b). The higher jitter in the join measurements when compared to time-of-flight and resulting higher variance of the fitted probability distribution, should be noted. These distributions were employed during delivered video quality experiments to simulated the behavior of the NICE application layer multicast protocol when used for multi-view video delivery. Although, not shown in the figures for visualization clarity, the undelivered packets are included in the time-of-flight measurements and they are taken into account by the fitted GMM distribution. There is approximately 6% chance of a packet being not delivered by the application layer multicast protocol. Although this figure is not insignificant, previous research suggests a lower delivery performance under frequent membership changes is to be expected for application layer multicast protocols.

### *3.3.3 Delivered Video Quality Performance*

In order to study the effects of the application layer multicast delivery network, we have used the join-latency and time-of-flight time models to generate simulated videos using the Race1 multi-view sequence from KDDI. These videos were generated using the same head trajectories as in Chapter 2 (Fig. 2.9(a) through Fig. 2.9(c)). First the head prediction was run on each of the head trajectories at various prediction distances. Then the videos, which would be seen by someone moving on a particular trajectory, are generated by issuing requests for predicted head position changes, causing a new join process and testing if actually required frame is available at its play-out deadline. In other words, the system requests the enhancement streams for the predicted head position but tries to use the high quality decoded picture for the actual head position. Naturally, the user is not moving abrupt enough to trigger stream changes at every frame, therefore time-of-flight model is used to test the availability of those frames at their play-out deadlines. In addition to availability, each frame is also tested for decodablity, for a frame to be decodable in low quality, it must be available and all the reference frames in the base layer MVC prediction

structure must be decodable as well. Additionally for a frame to be decodable in high quality, it needs to be available in the enhancement stream, decodable in low quality and its reference frames in the enhancement stream must be decodable as well. During test video generation the system first checks if the required frame at a particular time instance is decodable in high quality, if not it falls back to low quality, in the case the frame is not decodable in low quality as well, last displayed frame is repeated.

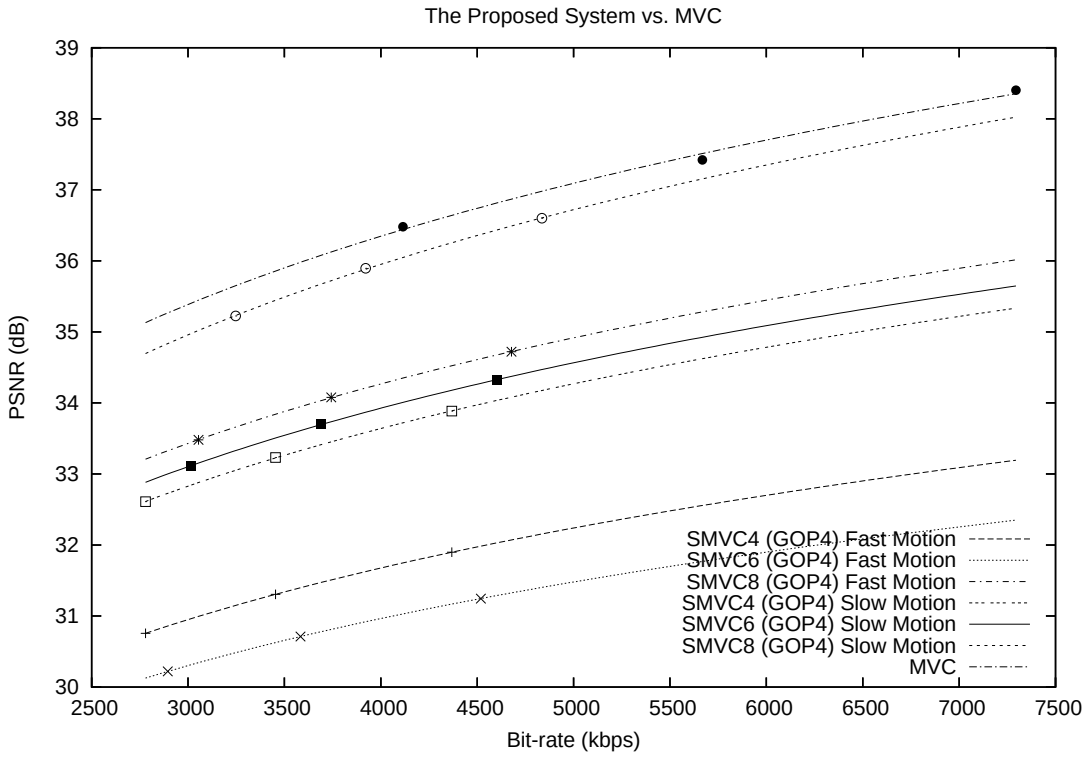
PSNR metric is computed from the generated test videos using reference videos for the same head trajectories. During the generation of a reference video for a head trajectory, the system assumes that all frames are always available in uncompressed from. Fig. 3.5(a) shows different rate-distortion curves for fast and slow head trajectories at three different numbers of views in the base layer: 4, 6 and 8. As it can be seen from the figure, the proposed system performs similar to MVC for slow head movements with 8 views in the base layer, however for other cases the performance disadvantage to the MVC is significant at the client side. This poor performance is largely due to the high number of undelivered frames due to multicast delivery tree instabilities, which arise from frequent node joins and leaves

### *3.3.4 Bandwidth Savings at the Server*

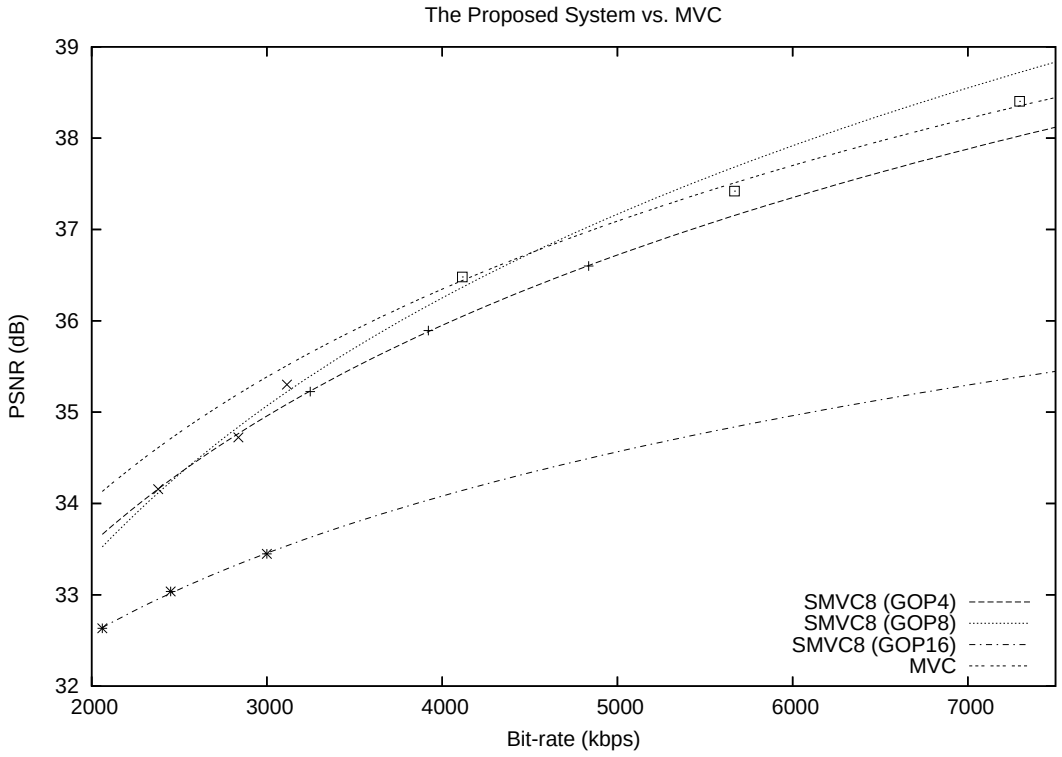
The performance of the proposed system is similar to sending the complete high quality MVC at the client-side. In order to asses the overall performance of the system the bandwidth savings at the server-side should be investigated as well. As it can be seen from Fig. 3.6

## *3.4 Discussion*

We have investigated how the proposed system with application layer multicast delivery trees performs under different head motion characteristics and GOP sizes. As it was shown in Section 3.3.3, although the proposed system matches the performance of MVC under suitable conditions, it suffers from the instabilities in the delivery trees,




(a) Rate-Distortion curves for fast and slow head motions with 4, 6 and 8 views in the base layer.




(b) Rate-Distortion curves for the slow head trajectory at different GOP sizes.

Figure 3.5: Two graphs showing how the performance of the proposed system changes

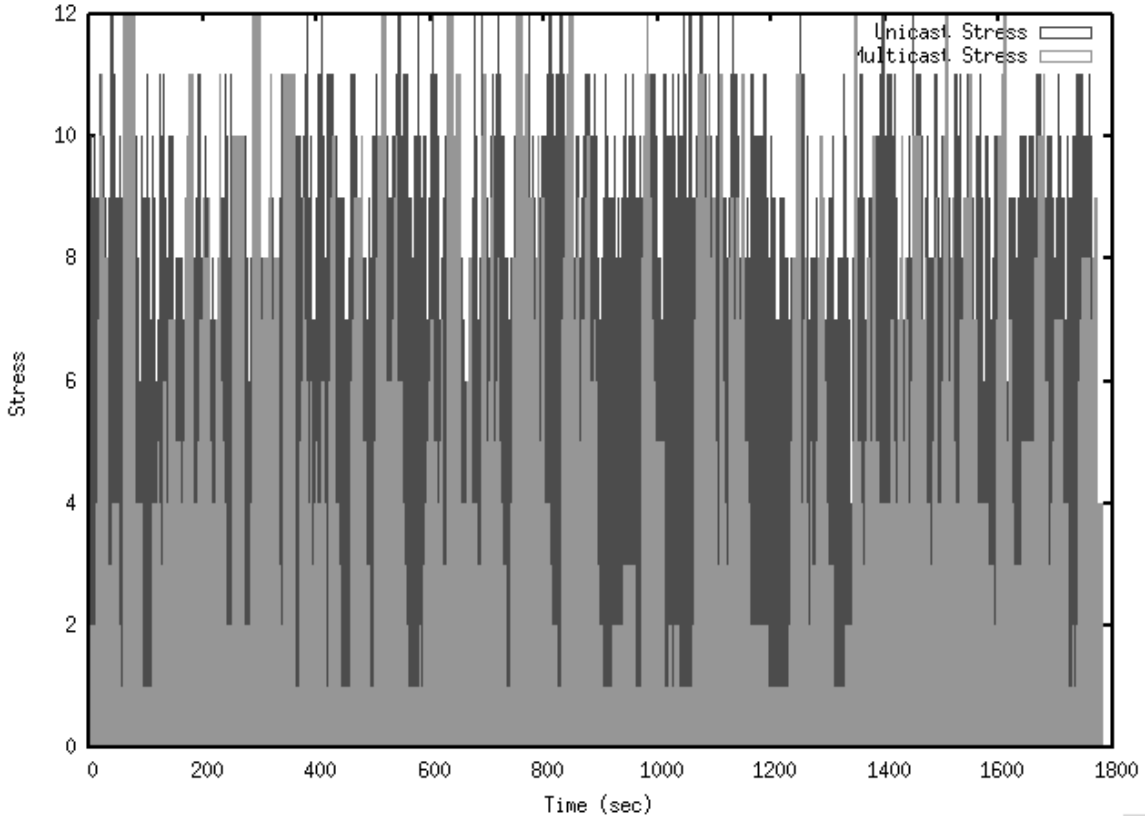



Figure 3.6: Change

which are caused by frequent join and leave actions of the peers triggered by the user's head motions. However, given the nature of current application layer multicast protocols, such frequent join and leave procedures are bound to cause unstable delivery trees and thus undelivered packets.

The important advantage of the proposed system is the reduction of stress at the server node provided by the application layer multicast architecture. Since the multicast peers distribute packets among themselves, the server only sends out packets to a few multicast peers, as opposed to sending packets to all clients in a unicast system. However, in order to make most of this advantage, the rate-distortion performance of the delivered video needs to be improved, such that the system overall provides more efficient use of bandwidth both at the server and at the client. To this end, the delivery performance of the application layer multicast protocol needs to be improved under frequent join and leave actions of peers.

# Chapter 4

# CONCLUSIONS

The main contribution of this thesis is the introduction of the selective delivery in two quality levels. As far as we are aware of, there has been no previous work on selectively delivering views from a multi-view video representation. We have shown that selectively delivering the required views, along with a few low quality views as a protection against prediction errors, can significantly outperform existing delivery methods for multi-view videos. Additionally we have also introduced a novel encoding structure which makes use of MVC and SVC concepts in order to achieve high compression rates while providing easy random access at the same time. Our results show that the proposed coding structure outperforms the reference system, especially at lower bit-rates.

This thesis also introduces delivery of multi-view video over an application layer multicast protocol. Although, multi-view video delivery over multicast results in lower rate-distortion performance at the client side, it reduces the bandwidth requirements and therefore costs at the server. Clearly, a client-driven multicast architecture for multi-view video delivery holds great potential. However, more further on application layer multicast protocols is needed in order to develop methods to reduce delivery tree instability during join or leave procedures. This would improve the packet delivery performance of the application layer multicast protocol, greatly improving the quality of the received multi-view video.

# BIBLIOGRAPHY

[1] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video th technologies, applications and MPEG standards," in *Proceedings of IEEE ICME 2006*. IEEE, 2006.

[2] C. Fehn, K. Hopf, and Q. Quante, "Key technologies for an advanced 3D-TV system," in *Proc. of SPIE Three-Dimensional TV, Video and Disp. III*, October 2004, pp. 66–80.

[3] C. Fehn, "Depth-Image-Based Rendering (DIBR), compression and transmission for a new approach on 3D-TV," *Proc. Stereoscopic Displays and Applications*, 2002.

[4] O. Scheer, C. Fehn, N. Atzpadin, M. Muller, A. Smolic, R. Tanger, and P. Kauff, "A flexible 3D TV system for different multi-baseline geometries," in *Proceedings of IEEE ICME 2006*. IEEE, 2006.

[5] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[6] M. Levoy and P. Hanrahan, "Light field rendering," in *SIGGRAPH '96*. New York, NY, USA: ACM Press, 1996, pp. 31–42.

[7] 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 (TOG)*, vol. 23, no. 3, pp. 814–824, 2004.

[8] K. Mueller, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, T. Oelbaum, and T. Wiegand, "Multi-view video coding based on H.264/AVC using hierarchical B-frames," in *Picture Coding Symposium 2006*. PCS, 2006.

[9] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, "A receiver-driven multicasting framework for 3DTV transmission," in *European Signal Processing Conference, Proceedings of*, September 2005.

[10] ——, "Interactive transport of multi-view videos for 3DTV applications," *Journal of Zhejiang University SCIENCE A: Proc. Packet Video Workshop 2006*, vol. 7, no. 5, pp. 830–836, 2006.

[11] R. Azuma and G. Bishop, "A frequency-domain analysis of head-motion prediction," in *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1995, pp. 401–408.

[12] J. Wu and M. Ouhyoung, "On latency compensation and its effects on head-motion trajectories in virtual environments," *The Visual Computer*, vol. 16, no. 2, pp. 79–90, 2000.

[13] A. Kiruluta, M. Eizenman, and S. Pasupathy, "Predictive head movement tracking using a kalman filter," *Systems, Man and Cybernetics, Part B, IEEE Trans. on*, vol. 27, no. 2, pp. 326–331, April 1997.

[14] L. Stelmach, W. Tam, D. Meegan, and A. Vincent, "Stereo image quality: effects of mixed spatio-temporal resolution," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 2, pp. 188–193, 2000.

[15] A. Aksay, C. Bilen, E. Kurutepe, T. Ozcelebi, G. B. Akar, M. R. Civanlar, and A. M. Tekalp, "Temporal and spatial scaling for stereoscopic video compression," in *EUSIPCO 2006*. EURASIP, 2006.

[16] E. H. Adelson and J. R. Bergen, *The Plenoptic Function and the Elements of Early Vision*. Cambridge, MA: MIT Press, 1991, pp. 3–20.

[17] S. B. Kang, R. Szeliski, and P. Anandan, "The geometry-image representation tradeoff for rendering," in *Image Proc., 2000. Proc. International Conference on*, vol. 2, 2000, pp. 13–16.

[18] C. Zhang and T. Chen, "A survey on image-based rendering–representation, sampling and compression," *Signal Proc. Image Comm.*, vol. 19, pp. 1–28, January 2004.

[19] X. Tong and R. M. Gray, "Interactive rendering from compressed light fields," *Circuits and Systems for Video Tech., IEEE Trans. on*, vol. 13, no. 11, pp. 1080–1091, November 2003.

[20] M. Magnor and B. Girod, "Data compression for light-field rendering," *Circuits and Systems for Video Tech., IEEE Trans. on*, vol. 10, no. 3, pp. 338–343, April 2000.

[21] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *SIGCOMM '02*. New York, NY, USA: ACM Press, 2002, pp. 205–217.

[22] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *SIGMETRICS '00*. New York, NY, USA: ACM Press, 2000, pp. 1–12.

[23] S. Banerjee, S. Lee, R. Braud, B. Bhattacharjee, and A. Srinivasan, "Scalable resilient media streaming," *Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*, pp. 4–9, 2004.

[24] E. Setton, J. Noh, and B. Girod, "Rate-distortion optimized video peer-to-peer multicast streaming," in *P2PMMS'05: Proceedings of the ACM workshop on*

*Advances in peer-to-peer multimedia streaming.* New York, NY, USA: ACM Press, 2005, pp. 39–48.

[25] M. Hosseini and N. D. Georganas, "Design of a multi-sender 3d videoconferencing application over an end system multicast protocol," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia.* New York, NY, USA: ACM Press, 2003, pp. 480–489.

[26] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *SIGCOMM '96.* New York, NY, USA: ACM Press, 1996, pp. 117–130.

[27] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *Proc. of the Fifth Symposium on Operating Systems Design and Implementation.* Boston, MA: USENIX Association, December 2002, pp. 255–270.

[28] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," in *SIGCOMM '02.* New York, NY, USA: ACM Press, 2002, pp. 133–145.

# VITA

ENGIN KURUTEPE was born in Istanbul, Turkey on February 21, 1982. He received his B.Sc. degrees in Electrical Engineering and Computer Engineering from Koç University, Istanbul, in 2004. From September 2004 to September 2006, he worked as a teaching and research assistant in Koç University, Turkey and conducted research on the transport of multi-view videos over IP-networks for 3DTV applications for the 3DTV NoE funded by the European Commission. He has published several chapters about on his research for the following conferences SIU2005 (Kayseyi, Turkey), EUSIPCO2005 (Antalya, Turkey), SIU2006 (Antalya, Turkey), PV2006 (Hangzhou, China), MRCS2006 (Istanbul, Turkey).