A Multi-stream Adaptation Framework for Bandwidth Management in 3D Tele-immersion

Zhenyu Yang, Bin Yu, Klara Nahrstedt
University of Illinois at Urbana-Champaign
Department of Computer Science
SC, 201 N. Goodwin, Urbana, IL 61801

Ruzena Bajscy
University of California at Berkeley
Department of EECS
253 Cory Hall, Berkeley, CA 94720

ABSTRACT

Tele-immersive environments will improve the state of collaboration among distributed participants. However, along with the promise a new set of challenges have emerged including the real-time acquisition, streaming and rendering of 3D scenes to convey a realistic sense of immersive spaces. Unlike 2D video conferencing, a 3D tele-immersive environment employs multiple 3D cameras to cover a much wider field of view, thus generating a very large volume of data that need to be carefully coordinated, organized, and synchronized for Internet transmission, rendering and display. This is a challenging task and a dynamic bandwidth management must be in place. To achieve this goal, we propose a multi-stream adaptation framework for bandwidth management in 3D tele-immersion. The adaptation framework relies on the hierarchy of mechanisms and services that exploits the semantic link of multiple 3D video streams in the tele-immersive environment. We implement a prototype of the framework that integrates semantic stream selection, content adaptation, and 3D data compression services with user preference. The experimental results have demonstrated that the framework shows a good quality of the resulting composite 3D rendered video in case of sufficient bandwidth, while it adapts individual 3D video streams in a coordinated and user-friendly fashion, and yields graceful quality degradation in case of low bandwidth availability.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]; C.2.1 [**Network Architecture and Design**]; H.5.1 [**Multimedia Information Systems**]: Video

General Terms

Design, Performance

Keywords

3D Tele-immersion, Bandwidth Management, Adaptation

NOSSDAV '06 Newport, Rhode Island USA

1. INTRODUCTION

The tele-immersive environments are emerging as the next generation technique for the tele-communication allowing geographically distributed users more effective collaboration in joint full-body activities than the traditional 2D video conferencing systems [16]. The strength of tele-immersion lies in its resources of a shared virtual space and the free-viewpoint stereo videos, which greatly enhance the immersive experience of each participant. Several early attempts [17, 5, 8] have illustrated the potential applications exemplified by virtual office, tele-medicine and remote education where an immersive collaboration is desirable. To advance the tele-immersive environments, ongoing research is carried out in areas of computer vision, graphics, data compression, and high-speed networking to deliver realistic 3D immersive experience in real time [15, 4, 8, 13, 9].

As pointed out in [18], one of the most critical challenges of tele-immersion systems lies in the transmission of multi-stream video over current Internet2. Unlike 2D systems, in a tele-immersive environment multiple cameras are deployed for wide field of view (FOV) and 3D reconstruction. Even for a moderate setting, the bandwidth requirements and demands on bandwidth management are tremendous. For example, the basic rate of one 3D stream may reach up to 100 Mbps and if considering 10 or more 3D cameras in a room the overall bandwidth could easily exceed Gbps level. To reduce the data rate, real-time 3D video compression schemes are proposed [10, 20] to exploit the spatial and temporal data redundancy of 3D streams.

In this paper, we explore the 3D multi-stream adaptation and bandwidth management for tele-immersion from the *semantic* angle. Although our work is motivated by the data rate issue, the idea is forged to address the concerns and challenges that are neglected by previous work. First, the multiple 3D streams are highly correlated as they are generated by cameras taking the same scene from different angles. The correlation is represented by not only the data redundancy but also the semantic relation among the streams. The semantic correlation demands an appropriate mechanism of coordination. Due to the absence of such a mechanism, most 3D tele-immersion systems handle all streams as equally important, resulting in low efficiency of resource usage. Second, although it is widely recognized that the interactivity through view selection is the key feature of 3D video applications [1], the feedback of user view does not play a central role in QoS control. As a consequence, current systems do not provide obvious way for a user to dynamically tune the quality according to his preference.

We address the data rate and bandwidth management issues utilizing the semantic link among multiple streams created due to the location and data dependencies among cameras, and interactive user preferences. The semantic link has not been fully used, developed and deployed for the purpose of the dynamic bandwidth management and high performance tele-immersion protocols over Internet networks in previous work. Hence, we propose to utilize the semantic link in the new multi-stream adaptation framework.

The design of the multi-stream adaptation framework revolves around the concept of *view-awareness* and a hierarchical service structure (Figure 1). The framework is divided into three levels. The *stream selection* level captures the user view changes and calculates the *contribution factor* (CF) of each stream for the stream selection. The *content adaptation* level uses a simple and fine-granularity method to select partial content to be transmitted according to the available bandwidth estimated by the underlying streaming control. The lowest *3D data* level performs 3D compression and decompression of the adapted data.

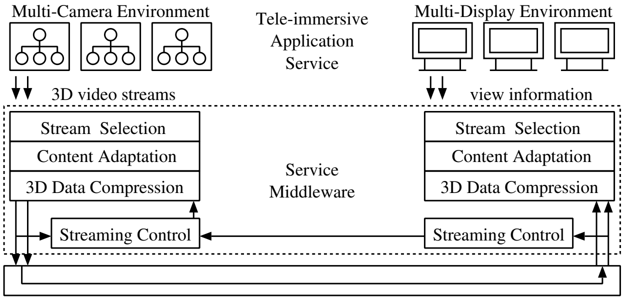


Figure 1: Hierarchical Multi-stream Adaptation Framework

The adaptation framework relies on several assumptions which are relevant to 3D tele-immersion. *A1.* Frequent view changes are desirable for applications in concern. *A2.* At any time, the user is only interested in one particular view. *A3.* Usually a wide field of view is covered (> 180 °) and the subject is not transparent (e.g., a person). Under these assumptions, the framework will differentiate among streams according to their contribution to the current view and select suitable adaptation per stream for dynamic bandwidth management and quality adjustment.

In summary, our hierarchical and semantic-driven 3D multi-stream adaptation framework for tele-immersive environments has the following contributions.

Framework. For the first time, we start to carefully consider the 3D multi-stream adaptation issue using the approach of a hierarchical framework which integrates user requirement, adaptation and compression.

View-awareness. The feedback of the user view becomes the centerpiece of the framework. This configuration matches the central role of the user view in tele-immersive applications. Therefore, the framework will navigate the adaptation and bandwidth management in a more intelligent way from the user's aspect.

Scalability. As in other distributed systems, the scalability in terms of the number of flows is a very critical issue. Otherwise, the networking and computational cost introduced by the adaptation may offset its benefit. Our adaptation scheme involves very small cost which makes it scalable in terms of the number of 3D streams.

We have implemented a prototype of the semantic protocol and the adaptation framework as part of the service middleware in the TEEVE project [21] to study the performance impact on the visual quality in both spatial and temporal terms. First, the rendered visual quality after the adaptation may degrade as compared with the case of no adaptation. Second, when the user changes his viewpoint there will be a certain end-to-end delay until the content based on the new view is streamed at full scale. We analyze the quality degradation through local and remote streaming experiments. The performance results have demonstrated that the adaptation framework achieves good rendering quality in case of sufficient bandwidth, while it dynamically adapts streams according to user preference and yields graceful quality degradation under bandwidth constraints.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents the TEEVE architecture. Section 4 explains the adaptation framework. Section 5 evaluates the performance. Section 6 concludes the paper.

2. RELATED WORK

We review previous work on multi-stream compression and adaptation algorithms for 3D videos.

2.1 3D Compression

The compression of 3D video streams (or *depth streams* since they contain the depth information) is a relatively new area. Pioneering work [10, 9] by Kum et al. has been focused on inter-stream compression scheme to save both networking and rendering cost. Under this scheme, the stream which is the closest to the user view is selected as the main stream while other streams are compared against it to remove redundant pixels that are within a threshold of distance. The major problems of inter-stream compression include (1) the considerable communication overhead as streams are initially distributed among different nodes and (2) the diminishing redundancy between streams that are not spatially close enough. To alleviate them, a group partitioning algorithm is applied. From the adaptation aspect, the scheme does take into account the current user view for selecting main stream. However, the group partitioning is a static process which does not dynamically adjust to the user view and all streams are treated as equally important. The distance threshold could serve as a tunable parameter but it is not straightforward from the user's perspective. Finally, the inter-stream compression faces a dilemma as the compression ratio is highly associated with the density of cameras. In one experimental setting of [9], 22 3D cameras are deployed with a horizontal field of 42 ° to achieve a 5 to 1 compression ratio. If the same setting is used to cover a much wider field of view, certain camera selection and 3D adaptation must be employed to scale the system.

The *multi-view video coding* (MVC) has recently become an active research topic including, for example, a multiview transcoder by Bai et al. [3], and ISO survey of MVC algorithms [2]. The common idea is to augment MPEG encoding scheme with cross-stream prediction to exploit temporal and spatial redundancy among different streams. However, as pointed out earlier the cross-stream compression could involve a very high communication overhead. Most implemented systems we have seen so far still encode each stream independently such as a multi-view video system by Lou et al. [11] and a 3D TV prototype by Matusik et al. [12].

As a contrast, the intra-stream compression schemes are proposed in [20], where each depth stream is independently compressed to remove spatial redundancy. Compared with the inter-stream compression, the intra-stream compression has better scalability as the number of streams increases and can be used in settings of sparse deployment of cameras, while achieving much higher compression ratio. However, the intra-stream compression does not reduce the number of pixels that need to be rendered. Therefore, it is necessary to apply an adaptation scheme to lower the rendering cost as the number of cameras increases. As shown later, the intra-stream compression can be easily integrated into our adaptation framework to compress the adapted data.

In summary, 3D adaptation is important under the context of available 3D compression techniques. Although the compression is one critical solution, it is not the complete answer to the end-to-end problem.

2.2 3D Adaptation

The 3D adaptation has been used in several tele-immersion systems and most of them take the view-based techniques due to two main reasons. First, currently the image-based approach is shown to be a feasible choice for real-time 3D video systems [1, 3], where cameras are densely distributed to reconstruct and render novel views from images taken in real scenes. Hence, this approach requires tremendous computational power, installation cost, storage and bandwidth if a large number of video streams are processed in full scale. Second, it is recognized that the interactivity through dynamic selection of viewpoints is the key feature of 3D video applications (as mentioned earlier).

Würmlin et al. implement a 3D video pipeline [14] for the blue-c telepresence project [6]. The video system installs 16 CCD cameras covering 360°. During the runtime, 3 cameras are selected for the texture and 5 cameras for reconstruction based on the user view. The concern of adaptation is more focused on the 3D video processing and encoding part to make it affordable within resource limitations. However, the issue of QoS adaptation according to the user requirement and available bandwidth, and the related spatial and temporal quality loss have not been addressed.

In other cases, the tolerance of human's perception is exploited to facilitate the design and implementation of 3D video systems. Ruigang et al. implement a prototype of the group video conferencing system [19], which uses a linear array of cameras mounted horizontally at the eye level to capture a compact light field as an approximation for light field rendering. However, no other adaptation scheme is applied and all cameras are selected. Hosseini et al. implement a multi-sender 3D videoconferencing system [7], where a certain 3D effect is created by placing the 2D stream of each participant in a virtual space. In their work, the adaptation is used to reduce the downlink traffic of each user based on the orientation of the view and its *visibility*. Conceptually, we borrow the similar idea but extend it into the 3D domain where each user is represented by multiple 3D streams.

3. ARCHITECTURE AND MODEL

To help the overall understanding of our multi-stream adaptation framework, we briefly present the overview of the TEEVE architecture and data model (details in [21]).

3.1 Architecture

The TEEVE architecture (Figure 1) consists of the *application* layer, the *service middleware* layer, and the underlying Internet transport layer. The application layer manipulates the multi-camera/display environment for end users including, for example, synchronizing 3D cameras for reconstruction, routing 3D streams onto multiple displays, and capturing user view changes. The service middleware layer contains a group of hierarchically organized services that reside within service gateways. These services explore semantic links among stream, content and user view information with respect to multiple cameras and displays. Based on the semantic link, they perform functions including multi-stream selection, content adaptation and 3D compression.

3.2 Model

There are N 3D cameras deployed at different viewpoints of a room. Each 3D camera i is a cluster of 4 calibrated 2D cameras connected to one PC to perform image-based stereo algorithm [13]. The output 3D frame f^i is a two dimensional array (e.g., 640×480) of pixels with each containing color and depth information. Every pixel can be independently rendered in a global 3D space, since its (x, y, z) coordinate can be restored by the row and column index of the array, the depth, and the camera parameters.

All cameras are synchronized via hotwires. At time t, the 3D camera array must have N 3D frames constituting a *macro-frame* F_t of $(f_t^1 \ldots f_t^N)$. Each 3D camera i produces a 4D stream S_i containing 3D frames $(f_{t_1}^i \ldots f_{t_\infty}^i)$. Hence, the tele-immersive application yields a 4D stream of macro-frames $(F_{t_1} \ldots F_{t_\infty})$.

4. ADAPTATION FRAMEWORK

Embedded in the service middleware (Figure 1), the adaptation framework includes the *stream selection*, *content adaptation* and *3D data compression*. Figure 3 gives a more detailed diagram of the framework. We concentrate on the stream and content levels, describing the protocol and related functions (details of the 3D compression in [20]).

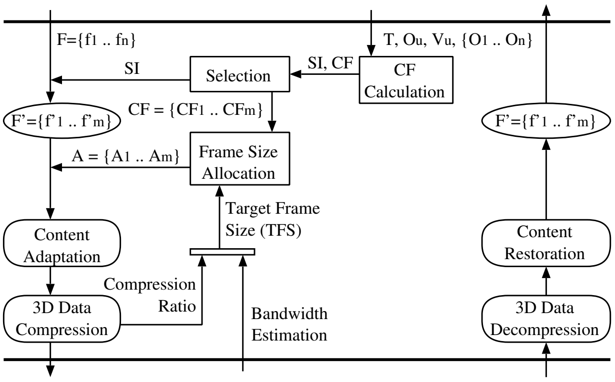
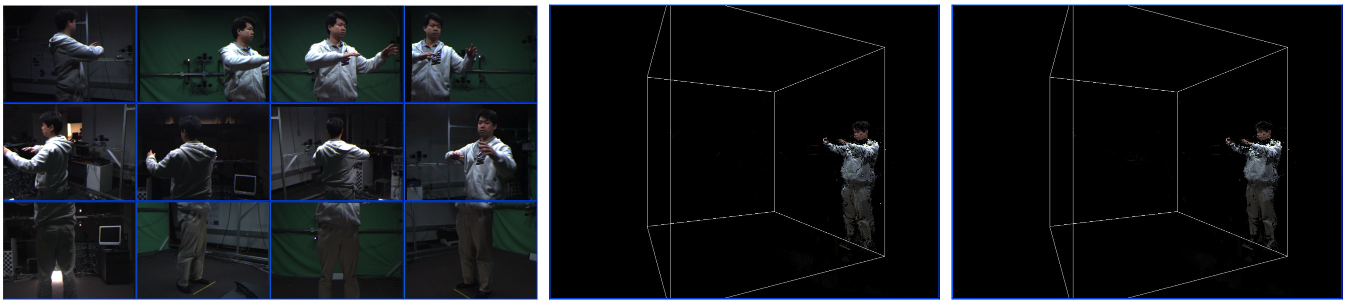


Figure 3: Adaptation Framework in Detail

4.1 Stream Selection Protocol

Step 1. Whenever the user changes his view, the information is captured at the receiver end, which triggers the *stream selection* and *CF calculation* functions. After that,



(a) Images from Multiple Cameras (b) 3D Rendering using 12 Cameras (c) 3D Rendering using 7 Cameras

Figure 2: Comparison of Visual Quality

the IDs of selected streams (SI) and associated contribution factors (CF) are transmitted to the sender end.

Step 2. The sender decides for each macro-frame F_t, the bandwidth allocation A_i of its individual 3D frames. The allocation is based on the user feedback of Step 1, the average compression ratio and the bandwidth estimation from the underlying streaming control.

Step 3. The bandwidth allocation is forwarded to the content adaptation level, where each stream is adapted, passed to the 3D data level for compression, and then transmitted.

4.2 Stream Selection

The orientation of camera i ($1 \leq i \leq N$) is given by the normal of its image plane, $\vec{O_i}$. The user view is represented by its orientation $\vec{O_u}$ and viewing volume V_u to capture view changes by rotation and translation. The user also specifies his preferable threshold of FOV as T ($T \in [0, 1]$). For unit vectors, the dot product $(\vec{O_i} \cdot \vec{O_u})$ gives the value of $cos\theta$, where θ is the angle between $\vec{O_i}$ and $\vec{O_u}$. When a camera turns away from the viewing direction of the user, its effective image resolution will decrease due to the foreshortening and occlusion. Thus, we use $(\vec{O_i} \cdot \vec{O_u})$ for the camera selection criterion and derive SI as in (1).

$$SI = \{i : (\vec{O_i} \cdot \vec{O_u}) \geq T, 1 \leq i \leq N\} \tag{1}$$

Figure 2 illustrates the stream selection effect. Figure 2a shows the color portion of 3D frames from 12 cameras. Figure 2b shows the 3D rendering effect when all cameras are used, while Figure 2c only uses the cameras by choosing $T = 0$ (i.e., a maximum of 90 ° from the viewing direction).

4.3 CF Calculation

The CF value indicates the importance of each selected stream depending on the orientation $\vec{O_u}$ and the volume V_u of the current user view. The viewing volume is a well-defined space within which objects are considered visible and rendered (*culling*). Given a 3D frame, we may compute the visibility of each pixel. To reduce the computational cost, we divide the image into 16×16 blocks and choose the block center as the reference point. The ratio of visible pixels is denoted as VR_i amd the CF is calculated in (2).

$$\forall i \in SI, \quad CF_i = (\vec{O_i} \cdot \vec{O_u}) \times VR_i \tag{2}$$

4.4 Frame Size Allocation

The goal of the streaming control is to keep the continuity of conferencing. For this, it maintains a stable frame rate while varying the macro-frame size to accommodate the bandwidth fluctuation. Based on the estimated bandwidth, the average compression ratio, and the desirable frame rate, the streaming control protocol suggests a *target macro-frame size* (TFS) to the upper level. Suppose the size of one 3D frame is fs. The task of the frame size allocation becomes critical when TFS is smaller than $m \times fs$ (where $m = |SI|$) and it has to choose a suitable frame size for each selected stream. We propose a *priority-based* allocation scheme which considers several factors. (1) Streams with bigger CF value should have higher priority. (2) Whenever possible, a minimum frame size defined as $fs \times CF_i$ should be granted. (3) Once (2) is satisfied, the priority should be given to cover a wider FOV.

We sort SI in descending order of CF to assign A_i. If $(TFS \geq fs \times \sum_{j \in SI} CF_j)$, the stream frame is allocated size as in (3),

$$A_i = min(fs, \; fs \times CF_i + \frac{(TFS - \sum_{j=1}^{i-1} A_j) \times CF_i}{\sum_{j=i}^{m} CF_j}) \tag{3}$$

which means after the minimum frame size is allocated, the residue of TFS is allocated proportional to CF. If $(TFS < fs \times \sum_{j \in SI} CF_j)$, then we allocate minimum stream frame size in order of priority (4).

$$A_i = min(fs \times CF_i, \; TFS - \sum_{j=1}^{i-1} A_j) \tag{4}$$

Thus, it is possible that some of the selected streams may not get the quota of transmission. To fully evaluate the performance, in later experiments we compare the priority scheme against the non-priority scheme (5).

$$A_i = \begin{cases} TFS/m & \text{if } TFS < m \times fs \\ fs & \text{otherwise} \end{cases} \tag{5}$$

4.5 Content Adaptation

The content adaptation layer adapts the 3D frame f_i for the assigned frame size A_i. As each pixel can be independently rendered, we take the approach of the pixel selection which provides a fine-granularity content adaptation. That is, we evenly select pixels according to the ratio of A_i/fs as we scan through the array of pixels. The ratio is attached to the frame header so that the row and column index of every selected pixel can be easily restored at the receiver end, which is needed for 3D rendering (Section 3).

5. PERFORMANCE EVALUATION

We embed the adaptation framework in the TEEVE service middleware. For evaluation, we use 12 3D video streams (320 × 240 resolution) pre-recorded from the multi-camera environment showing a person and his physical movement with a horizontal FOV of 360 °.

5.1 Overall Rendering Quality

The first set of experiments are performed on the local testbed, where we send video streams to the 3D renderer within the same Gigabit ethernet. The adaptation is configured to choose TFS between 8 fs and 1 fs. Meanwhile, we gradually rotate and shift the view during the experiment. Figure 4 shows the comparison of the rendered quality of the two schemes. The peak signal-to-noise ratio (PSNR) is calculated by comparing with the base case of full streaming (i.e., 12 streams each with 100% content).

When TFS is large, the two schemes have the same quality because each selected stream can be transmitted with the full content. When TFS is further reduced (macro-frame number > 500 in Figure 4), the two schemes show different quality degradation. For most of the cases, the quality of priority scheme is better than the non-priority scheme. The trend continues until TFS is reduced to around 2 fs (macro-frame number > 1300). Then the qualities mix with each other. In the priority scheme, only part of the body is rendered because some of the streams are dropped. However, in the non-priority scheme, the full body is still visible. The average PSNR is shown in Table 1.

5.2 Rendering Time

The renderer is implemented with OpenGL. We measure the rendering time using a Dell Precision 470 computer with 1 GByte memory running Windows. The average rendering time of 12 streams is 159.5 ms per macro-frame. Table 1 shows the average rendering time for each TFS (we combine the results of both schemes as they are very similar).

TFS (fs)	Average PSNR		Time
	Priority	Non-Pri.	
7	39.75	39.38	93.83
6	36.85	33.31	84.63
5	34.46	31.48	68.36
4	31.79	30.02	55.87
3	30.15	28.98	43.82
2	27.71	27.66	31.89
1	26.42	26.91	22.59

Table 1: PSNR (dB) and Rendering Time (ms)

5.3 Rendering Quality of View Changes

One important consequence of the multi-stream adaptation is the delay of response. When the user changes his view, the SI and CF will change accordingly which requires streams of new configuration to be transmitted so that the new view can be correctly rendered. We set up a remote streaming testbed between University of Illinois at Urbana-Champaign and University of California at Berkeley to study the temporal impact of adaptation.

We select $TFS = 5$ fs and stream the 3D videos at the frame rate of 10 Hz. The renderer has the buffer space of 1 macro-frame. The average round-trip transmission delay is 86 ms, which is the duration between the renderer sends a request and the new macro-frame arrives. Thus, we keep the end-to-end delay below 200 ms. Figure 5 shows part of the streaming results which illustrate the quality degradation following view changes. Overall, the degradation of the priority scheme is bigger than the degradation of the non-priority scheme, especially when we apply large view changes. However, for both schemes the quality improves within the next two or three frames. The average delay between the time when the user made the view change and the time when the new view is correctly rendered is 237 ms.

6. CONCLUSION

We present a multi-stream adaptation framework for bandwidth management in 3D tele-immersive environments. The framework features a hierarchical structure of services and takes the user view and the semantic link among streams, content and compression into account. The semantic information guides the stream selection, the content adaptation and the 3D data compression at different levels of the end-to-end system. We propose a criterion, the contribution factor, for differentiating the importance of each 3D stream. Based on the *CF*, we utilize a priority scheme for bandwidth management. We compare the rendering quality of our approach with the adaptation of a non-priority scheme in both local and remote streaming tests while varying the target macro-frame size and applying different levels of view changes. Under small and gradual view changes, the priority scheme achieves better rendering quality for most cases. When the view changes increase, the quality degradation of the priority scheme becomes higher within a short interval of two or three frames.

For the future work, we are interested in considering the scenario of multiple views at the receiver end, investigating other content adaptation techniques, and developing quality prediction mechanisms for adapting 3D videos.

7. ACKNOWLEDGEMENT

We would like to acknowledge the support of this research by the National Science Foundation (NSF SCI 05-49242, NSF CNS 05-20182). The presented views are those of authors and do not represent the position of NSF. We would also like to thank Sang-hack Jung for providing us the tele-immersive videos.

8. REFERENCES

[1] Report on 3dav exploration. *International Organisation for Standardisation, ISO/IEC JTC1/SC29/WG11 N5878*, July 2003.

[2] Survey of algorithms used for multi-view video coding. *International Organisation for Standardisation, ISO/IEC JTC1/SC29/WG11 N6909*, January 2005.

[3] B. Bai and J. Harms. A multiview video transcoder. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 503–506, New York, NY, USA, 2005. ACM Press.

[4] H. Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, M. Goss, W. Culbertson, and T. Malzbender. Understanding performance in coliseum, an immersive videoconferencing system. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1, 2005.

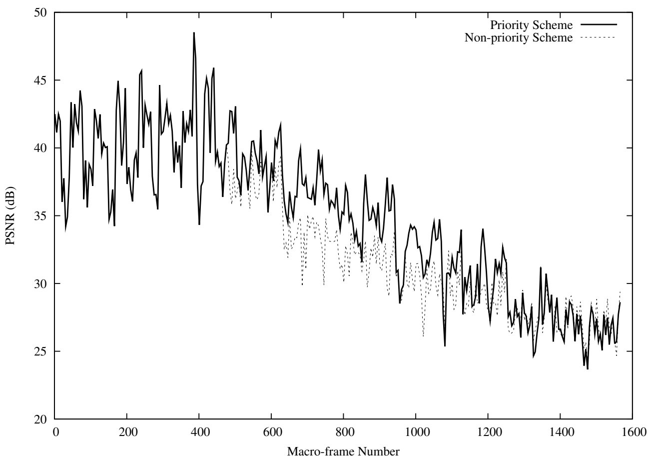


Figure 4: Overall Rendered Quality

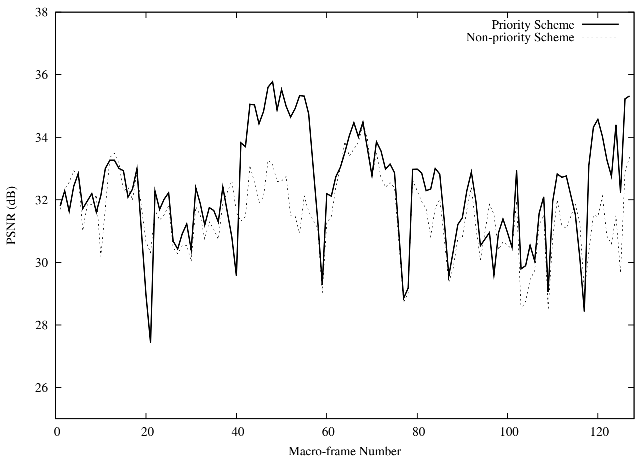


Figure 5: Quality of View Change

[5] K. Daniilidis, J. Mulligan, R. McKendall, D. Schmid, G. Kamberova, and R. Bajcsy. Real-time 3d-teleimmersion. In *Confluence of Computer Vision and Computer Graphics*, pages 253–265, 2000.

[6] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. blue-c: a spatially immersive display and 3d video portal for telepresence. *ACM Trans. Graph.*, 22(3):819–827, 2003.

[7] 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*, pages 480–489, New York, NY, USA, 2003. ACM Press.

[8] P. Kauff and O. Schreer. An immersive 3d video-conferencing system using shared virtual team user environments. In *CVE '02: Proceedings of the 4th international conference on Collaborative virtual environments*, pages 105–112, New York, NY, USA, 2002. ACM Press.

[9] S.-U. Kum and K. Mayer-Patel. Real-time multidepth stream compression. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(2):128–150, 2005.

[10] S.-U. Kum, K. Mayer-Patel, and H. Fuchs. Real-time compression for dynamic 3d environments. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 185–194, New York, NY, USA, 2003. ACM Press.

[11] J.-G. Lou, H. Cai, and J. Li. A real-time interactive multi-view video system. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 161–170, New York, NY, USA, 2005. ACM Press.

[12] W. Matusik and H. Pfister. 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Trans. Graph.*, 23(3):814–824, 2004.

[13] J. Mulligan and K. Daniilidis. Real time trinocular stereo for tele-immersion. In *International Conference on Image Processing*, pages III: 959–962, 2001.

[14] S. Mürmlin, E. Lamboray, and M. Gross. 3d video fragmens: dynamic point samples for real-time free-viewpoint video. In *Technical Report No. 397*, Institute of Scientific Computing, ETH, Zurich, 2003.

[15] D. E. Ott and K. Mayer-Patel. Coordinated multi-streaming for 3d tele-immersion. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 596–603, New York, NY, USA, 2004. ACM Press.

[16] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 179–188, New York, NY, USA, 1998. ACM Press.

[17] H. Towles, W.-C. Chen, R. Yang, S.-U. Kum, and H. F. et al. 3d tele-collaboration over internet2. In *International Workshop on Immersive Telepresence (ITP 2002)*, December 2002.

[18] H. Towles, S.-U. Kum, T. Sparks, S. Sinha, S. Larsen, and N. Beddes. Transport and rendering challenges of multi-stream, 3d tele-immersion data. In *NSF Lake Tahoe Workshop on Collaborative Virtual Reality and Visualization*, Octomber 2003.

[19] R. Yang, C. Kurashima, A. Nashel, H. Towles, A. Lastra, and H. Fuchs. Creating adaptive views for group video teleconferencing - an image-based approach. In *International Workshop on Immersive Telepresence (ITP 2002)*, 2002.

[20] Z. Yang, Y. Cui, Z. Anwar, R. Bocchino, N. Kiyanclar, K. Nahrstedt, R. H. Campbell, and W. Yurcik. Real-time 3d video compression for tele-immersive environments. In *SPIE Multimedia Computing and Networking (MMCN 2006), San Jose, CA*, January 2006.

[21] Z. Yang, K. Nahrstedt, Y. Cui, B. Yu, J. Liang, S. hack Jung, and R. Bajcsy. Teeve: The next generation architecture for tele-immersive environments. In *IEEE International Symposium on Multimedia (ISM2005)*, Irvine, CA, USA, 2005.