A Multi-stream Adaptation Framework for Tele-immersion

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

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

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. 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. However, one of the most critical challenges of tele-immersion systems lies in the transmission of multi-stream video over current Internet infrastructure. Unlike 2D systems, a tele-immersive environment employs multiple cameras 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 [3, 5] to exploit the spatial and temporal data redundancy of 3D streams.

In this paper, we explore the multi-stream adaptation and bandwidth management for 3D tele-immersion. Although our work is majorly motivated by the data rate issue, the idea of multi-stream adaptation is forged to address the concerns and challenges that are neglected by previous research work. First, the multiple 3D video streams are highly correlated as they are produced by cameras taking the same scene from different viewpoints. There is a strong semantic link among the 3D cameras and the user views. The correlation demands an appropriate mechanism of multi-stream coordination for the purpose of QoS adaptation. Due to the absence of such mechanism, most 3D tele-immersion sytems handles all streams as equally important resulting in low efficiency of resource usage. Second, It is widely recognized that the interactivity through the free view selection is the key feature of 3D video applications. However, the feedback of user view does not play a central role in the control of media transmission.

We address the data rate and bandwidth management issues utilizing the semantic link among multiple streams created due to the interaction of camera orientations and the user view changes. The semantic link has not been fully developed and deployed for the purpose of the dynamic bandwidth management and high performance tele-immersion protocols over the Internet 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

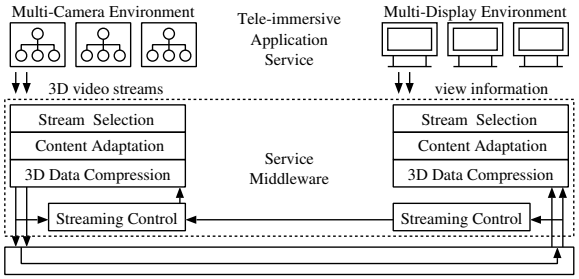


Figure 1: Multi-stream Adaptation Framework

the user view changes and calculates the *contribution factor* (CF) of each stream. 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.

Copyright is held by the author/owner(s).
MM'06, October 23–27, 2006, Santa Barbara, California, USA.
ACM 1-59593-447-2/06/0010.

2. ARCHITECTURE AND MODEL

The adaptation framework is embedded as part of the service middleware in our tele-immersion project known as TEEVE. We briefly present the overview of the TEEVE architecture and data model (more details in [6]).

2.1 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. The service middleware layer contains a group of hierarchically organized services that reside within service gateways. These services explore semantic links derived from the camera orientation and the user view information. Based on the semantic link, they perform functions including multi-stream selection and content adaptation.

2.2 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 trinocular stereo algorithm. 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 $(f_t^1 \ ... \ f_t^N)$. Each 3D camera i produces a 4D stream S_i containing 3D frames $(f_{t_1}^i \ ... \ f_{t_\infty}^i)$. Hence, the tele-immersive application yields a 4D stream of macro-frames $(F_{t_1} \ ... \ F_{t_\infty})$.

3. ADAPTATION FRAMEWORK

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 (details of 3D compression are in [5]).

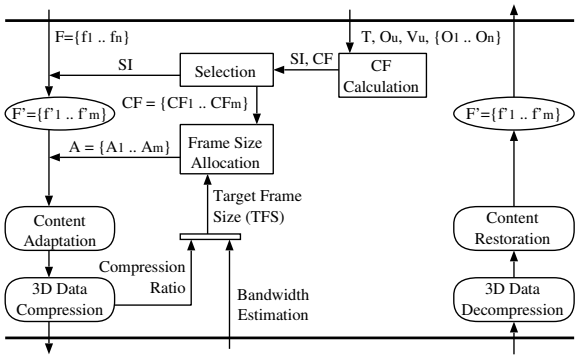


Figure 3: Adaptation Framework in Detail

3.1 Stream Selection Protocol

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

3.2 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. 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).

3.3 CF Calculation

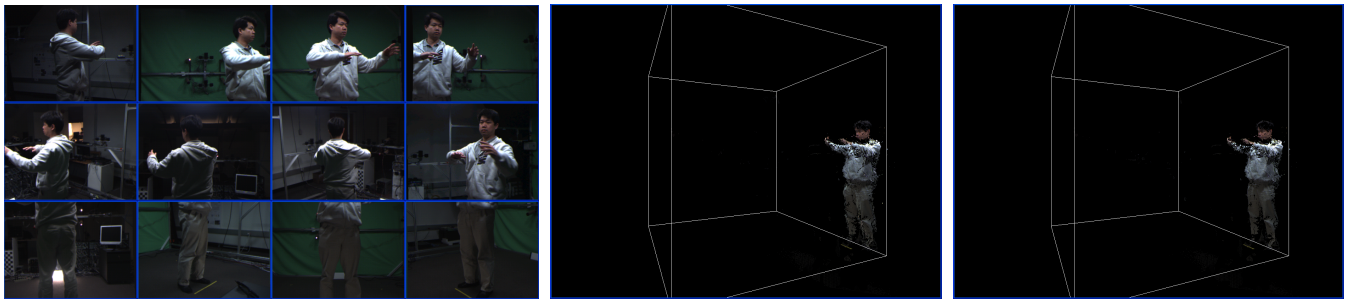
The CF value indicates the importance of each selected stream depending on the orientation $\vec{O_u}$ and the volume V_u of the current user view. The viewing volume is a well-defined space within which objects are considered visible and rendered (*culling*). Given a 3D frame, we may compute the visibility 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 and the CF is calculated in (2).

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

3.4 Frame Size Allocation

The streaming control stabilizes the 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 is to determine a suitable frame size for each selected stream. We propose a *priority-based* allocation scheme with the following principles. (1) Streams with bigger CF value should have higher priority. (2) When 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 $(TFS \geq fs \times \sum_{i \in SI} CF_i)$, the stream frame is allocated



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

Figure 2: Comparison of Visual Quality

size as in (3) where $m = |SI|$.

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

If $(TFS < fs \times \sum_{i \in SI} CF_i)$, then we allocate minimum stream frame size in order of priority (4).

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

Thus, it is possible that some of the selected streams may not get the quota of transmission.

3.5 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 2).

4. RELATED WORK

We review previous work on multi-stream compression and adaptation algorithms for real-time tele-immersion systems. In [3], Kum et al. present the inter-stream compression where multiple streams are compared to remove redundant pixels. In [1], the multi-view video coding is proposed to augment MPEG encoding scheme with cross-stream prediction to exploit temporal and spatial redundancy among different streams. In [5], Yang et al. propose the intra-stream compression where each video stream is independently compressed to remove spatial redundancy. As a contrast, we focus on the multi-stream adaptation.

In [4], Mürmlin et al. implement a 3D video pipeline for the blue-c telepresence project. During the runtime, cameras are selected for the texture and 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. Hosseini et al. implement a multi-sender 3D videoconferencing system [2], where a certain 3D effect is achieved by placing the 2D stream of each participant in a virtual space. Conceptually, we borrow the similar idea but extend it into the 3D domain where each user is represented by multiple 3D streams.

5. CONCLUSION

In this paper, 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. 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.

6. ACKNOWLEDGEMENT

The research is supported by the National Science Foundation (NSF SCI 05-49242, NSF CNS 05-20182), but the views are those of authors.

7. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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*, Sanjose, CA, USA, 2006.

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