Experience with Multi-Camera Tele-Immersive Environment

Jin Liang, Zhenyu Yang, Bin Yu, Yi Cui, Klara Nahrstedt
University of Illinois at Urbana-Champaign
{jinliang, zyang2,binyu,yicui,klara}@cs.uiuc.edu
Sang-Hack Jung, Art Yeap, Ruzena Bajcsy
University of California at Berkeley
{sangj,arty,bajcsy}@eecs.berkeley.edu

1. INTRODUCTION

With their ability to extract, transfer and render 3D models of real world objects, multi-camera, tele-immersive environments are becoming promising next-generation of tele-communication systems. By rendering the 3D model of a remote object in an arbitrary way, determined by each individual viewer, tele-immersive environments offer great richness in tele-communication that is lacking in existing 2D based systems.

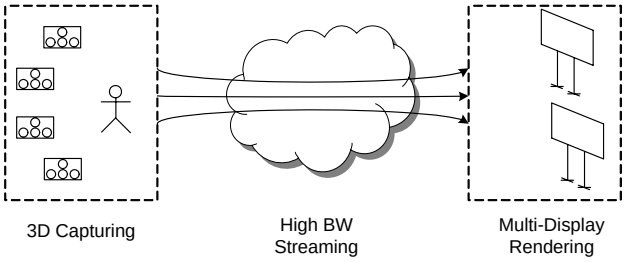

Figure 1. ***Multi-Camera, Multi-Display Tele-Immersive System***

Figure 1 illustrates the tele-immersive system, called TEEVE (TEle-immersive Environment for Everybody), we are building between University of Illinois at Urbana-Champaign (UIUC) and University of California at Berkeley (UCB) (see also the infrastructures in Figure 2 and 3) . At the sending site, multiple camera clusters are used to capture the 3D model (depth as well as color information) of real world objects in real time. The 3D models in form of multiple video streams are streamed over the Internet and rendered on multiple display devices at the receiving site. Figure 1 shows only one direction in the communication. In reality, both of the sites are capable of capturing and rendering the 3D models.

Despite their potential, building 3D-based tele-immersive systems faces numerous challenges, due to the high-fidelity requirement on 3D reconstruction and the lack of existing hardware and software tools. Specifically, (1) the basic 3D camera cluster must be *constructed by hand*, due to the lack of existing off-the-shelf commercial products; (2) the setup of multiple camera clusters must be *custom-designed*, with the particular communication application (e.g.,

Figure 2: Early *TEEVE Setup at UIUC*

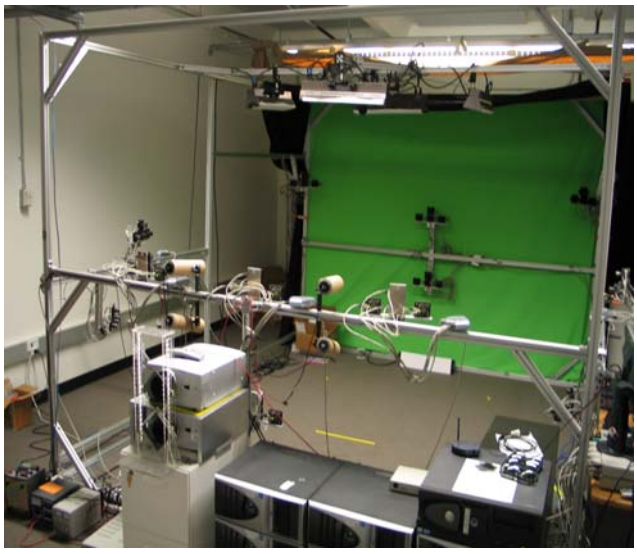

Figure 3: ***TEEVE Setup at UCB***

conferencing or distributed dancing choreography) in mind; (3) the multiple cameras must be *accurately calibrated*, in order to obtain high quality 3D models; (4) the high bandwidth 3D data must be *streamed across the best-effort Internet in a timely fashion*, in order to facilitate interactive communications. In this extended abstract, we describe our experiences in addressing these challenges and the lessons learned. We first describe the issues related to hardware and software setup, and then present briefly

lessons learned with respect to the bandwidth management scheme for 3D video streaming. Finally, we provide some concluding remarks.

2. HARDWARE SETUP EXPERIENCES

2.1 3D Camera Clusters

3D camera clusters are the basic units for the 3D reconstruction. Figure 4 shows the camera cluster used in our tele-immersive environment. Each cluster consists of four Point Grey Research DragonFly™ [1] cameras. Three of the cameras are black and white, and are used to extract the depth information of pixels using trinocular [2] stereo algorithms. The fourth camera is a color camera and is used to obtain the RGB color information of each pixel. Each camera cluster can produce a 3D video stream from one viewpoint.

Figure 4. ***3D Camera Cluster***

The camera clusters are built by hand mounting the four cameras on aluminum mounts. Our mounts have been constructed at the university machine workshop. However, our experience is that when designing the 3D camera clusters, it must be considered how the 3D camera cluster is going to be mounted on the metal framework that holds the multiple cameras. The 3D cameras need to use compatible nuts and bolts. Also, when setting up the camera clusters, it is desirable that the clusters can be tilted so that different applications can be deployed in TEEVE (e.g., conferencing or dancing). Thus compatible holding devices such as vives must also be considered.

2.2 Firewire Cards and Hubs

Each camera cluster is connected to one computer. Since each camera needs an IEEE 1394 (firewire) interface, and one computer may not provide enough firewire ports, we use firewire hubs to connect the cameras. The cameras are first connected to the hub, which is then connected to the computer. Using firewire hubs has the additional benefit that if the camera cluster is far from its computer, only one long firewire cable is needed.

Since all four cameras of a cluster share a single port on a firewire PCI card, the performance of the PCI card may become a bottleneck when high frame rate images need to be grabbed from the cameras. We have learned that some existing cards cannot support all four cameras at full frame rate (15 frames per second). Instead, the Lucent/Agere firewire cards can provide the necessary I/O performance since they have as many as 8 DMA channels.

2.3 Multiple 3D Camera Clusters Configuration

The configuration of camera clusters is important to the 3D reconstruction. The particular configuration depends on the application and the room environment. For example, whether full body or half body 3D models need to be reconstructed, how big is the scene area, whether eye-contact needs to be considered, and other issues. However, there are still some guidelines. First, the clusters are often mounted on metal frameworks. It is important that the frameworks are steady and flexible. Second, for any adjacent camera clusters, there must be overlap in their reconstructed 3D models. This is to ensure that the real world objects are completely covered. However, the overlap should not be too large. Otherwise, there will be too much redundancy in the 3D models. Third, lighting condition is important in a tele-immersive environment. Ideally, the light should come from all around the objects, instead from the room sealing only. However, it should be avoided that some lights are directly visible to the cameras. Also, florescent lights should be avoided, due to the flickering in the light, whose frequency is close to the frame rate that the cameras work. Our current setting still uses florescent lights. We are considering moving to other kind of lights.

2.4 Infra-red Cameras

The trinocular stereo algorithm for 3D reconstruction is based on point correspondence. This means given a physical point, the algorithm must search for the corresponding image point in different pictures. To improve the accuracy of point matching, infrared cameras and their lights can be used as shown in Figure 3. These lights can shed random patterns on the physical object. These patterns are visible to the black and white cameras. Therefore, they make it easier to search for matching points. On the other hand, infrared light is filtered in the color camera; therefore the patterns are not visible in the color information.

Figure 5: ***Infra-red Camera***

2.5 Trigger Wires

The 3D models reconstructed by different camera clusters are rendered at the same time at the receiving site. Therefore, it is important that different camera clusters grab images at the same time as well as that the network preserves the timing synchrony. This is achieved at the

sending site by hardware synchronization and at the receiving site by synchronization approaches embedded in the end-to-end protocols. We discuss our experience with the hardware synchronization. Specifically, each DragonFly camera can be configured in an external trigger mode. In the trigger mode, the camera will grab an image only when a trigger pulse is given on two trigger pins. Trigger pulse can be produced by writing 0s and 1s to the parallel port, which is connected to the trigger pin of the cameras. For load balancing purpose, it is better each parallel port pin is connected to equal number of cameras.

3. SOFTWARE CALIBRATION EXPERIENCE

3.1 Camera Parameter Setting

Before the cameras can work, they must be manually tuned to adjust their focus and aperture. After that, we still need to set the correct parameters such as brightness, shutter, gain, etc. Our Berkeley partners have developed image processing and calibration software tools to tune the parameters in a visual way. This greatly speeds up the process for parameter setting.

3.2 Camera Calibration

To reconstruct 3D information of real world objects, the cameras must be calibrated. This means the intrinsic (e.g., focal length, principal point) and extrinsic (e.g. translation and rotation) of the cameras must be calculated. Calibrating large number of cameras (we have 40 cameras right now) is a very difficult task. Existing tools such as the Bouguet [3] toolbox are designed for small number of cameras. Research groups at University of Pennsylvania and University of California at Berkeley are designing new calibration tools. The basic idea is to take grid pattern pictures on each camera, and let the calibration tool automatically extract the grid corners and compute the intrinsic parameter of the cameras. To compute the extrinsic parameter, different cameras take the picture of a point light source at the same time. The bright dot in each picture is then extracted and analyzed.

Although the new tool greatly facilitates multi-camera calibration, it still involves a lot of work. First, we need to take a series of grid pattern pictures on each camera. Our Berkeley partners have developed a tool for taking pictures on multiple cameras at the same time. We have also developed a tool that displays the grid patterns on each camera while taking the pictures. This avoids taking pictures with incomplete grid patterns. Second, we need to take a large number of dot light pictures on different cameras. This means we must make the room completely dark and display only a point light source (e.g., a small flash light bulb) while taking the pictures (hardware synchronized). To improve the accuracy of extrinsic parameter computation, it is better that thousands of pictures on each camera are taken. Third, after the grid pattern and dot light pictures are taken, we must manually calibrate two reference cameras (using the Bouguet tool box), and then run through the new calibration tool to calibrate the rest cameras.

4. BANDWIDTH MANAGEMENT

Each camera cluster produces a 3D video stream that has more than 3MB raw data per second. To transmit the high bandwidth data across the Internet2, we need to have in place many algorithms and protocols such as (a) appropriate compression algorithms of 3D video data, (b) end-to-end video protocols that provide appropriate traffic shaping, preserve time synchronization and work with user, system and network feedback, (c) distributed resource management, (d) multi-stage feedback approaches, and many others. We have gathered initial experience with two issues, the 3D compression and the end-to-end streaming protocol. First, we have implemented and experimented with a 3D compression algorithm. This algorithm separates the 3D video model into two parts, the depth and color information and we compress the depth using lossless coding (e.g., Run-length coding) and color using lossy coding (e.g., Motion JPEG). Second, we have implemented and experimented with a simple end-to-end 3D video protocol that uses intermediate gateway controllers to serialize and shape the transmission of multi-camera video data[1]. This avoids the burstiness that is caused by different computers, sending the data at exactly at the same time.

5. CONCLUSION

In summary, building tele-immersive environments such as TEEVE at UIUC and UCB (see Figure 4 and Figure 5) still requires significant amount of time and effort with today's technology . There are several reasons. First, many hardware components such as camera clusters and camera framework are not readily available and must be hand made. Second, the camera setup depends largely on the particular application and room environment. Third, each cluster is connected to a computer, and the computers are not easily managed. Fourth, existing software tools are not sufficiently automated.

To facilitate the setup of similar systems, we believe (1) example hardware specifications for cameras, mounts, frameworks, computers, firewire cards, etc. should be given, so that they can be made/purchased more easily; (2) guidelines for camera setup, including parameter setting, cluster positioning and lighting condition should be provided; (3) software tools such as those for viewing

[1] This is a paradox, since at the source camera site hardware synchronization occurs first, but then for Internet2 transmission we must serialize the 3D video data, and introduce shaping delays to get data through the network and not overwhelm the system resources with the large amount of data, and again the serialized data needs to be re-synchronized at the receiving site for rendering/display purposes.

pictures across computers, taking simultaneous pictures in a visual way would be invaluable to the system setup and (4) software tools and protocols should be in place for managing multiple computers and delivering ten and more streams in a synchronized fashion. Thus great efforts must be put into designing new tools, protocols, and integrating existing tools into the overall software system framework.

6. REFERENCES

[1] Point Grey Research, http://www.ptgrey.com/, 2005

[2] J. Mulligan and K. Daniilidis. *Real time trinocular stereo for tele-immersion*. In International Conference on Image Processing, pages III: 959–962, 20

[3] Calibration toolbox, http://www.vision.caltech.edu/bouguetj/calib_doc/, 2005