REMOTE AND COLLABORATIVE 3D INTERACTIONS

Benjamin Petit[*], *Jean-Denis Lesage*[*], *Edmond Boyer*[*], *Jean-Sébastien Franco*[†], *Bruno Raffin*[*]

ABSTRACT

We present a framework for new 3D tele-immersion applications that allows collaborative and remote 3D interactions. This framework is based on a multiple-camera platform that builds, in real-time, 3D models of users. Such models are embedded into a shared virtual environment where they can interact with other users or purely virtual objects. 3D models encode geometric information that is plugged into a physical simulation for interactive purposes. They also encode photometric information through the use of mapped textures to ensure a good sense of presence. Experiments were conducted with two multiple-camera platforms, and the preliminary results demonstrate the feasibility of such environments.

***Index Terms*—** Tele-Immersion; Telepresence; Collaborative 3D Interactions; Markerless 3D Modeling; Multi-camera

1. INTRODUCTION

Tele-immersion is of central importance for the next generation of live and interactive 3DTV applications. It refers to the ability to embed persons at different locations into a shared virtual environment. In such environments it is essential to provide users with a credible sense of 3D telepresence and interaction. Several technologies already offer 3D experiences of real scenes with 3D and free-viewpoint visualization. However, live 3D tele-immersion and interactions across remote sites, is still a largely challenging goal. This is not only because of the intrinsic difficulty of building live 3D models that are compact, transfer friendly and visually rich. Indeed it is also necessary to ensure that these models can be used with the physical simulations needed for interaction with virtual objects. In this paper, we address all of these issues and propose a complete framework allowing the full immersion of distant people into a single collaborative and interactive environment.

The interest of virtual immersive and collaborative environments arises in a large and diverse set of application domains, including interactive 3DTV broadcasting, video gaming, social networking, 3D teleconferencing, collaborative manipulation of CAD models for architectural and industrial processes, remote learning, training, and other collaborative tasks such as civil infrastructure or crisis management. Such environments strongly depend on their ability to build a virtualized representation of the scene of interest, e.g. a 3D model of a user. Most existing systems use 2D representations obtained using mono-camera systems. While giving a partially faithful representation of the user, they do not allow for natural interactions, including consistent visualization with occlusions, which require full 3D descriptions. Other systems more suitable for 3D virtual worlds use avatars, as for instance massively multi-player games analog to *Second Life*. However, avatars only carry partial information about users and although real-time motion capture environments can improve such models and allow for animation, avatars do not yet provide sufficiently realistic representations for tele-immersive purposes.

To improve the sense of presence and realism, models with both photometric and geometric information should be considered. They yield more life-like representations that include user appearance, motions and even facial expressions. Different strategies can be followed to obtain such 3D models of a person. Multi-camera systems are often considered for that purpose and can provide either 2D + depth or full 3D geometric models. 2D + depth approaches provide a viewpoint dependent and thus incomplete 3D data. It enables to some extent free-viewpoint 3D visualization, but interactions are limited. Full 3D approaches are already used for tele-immersion [1] and they are more suitable for interactions as they yield more information. Nevertheless, existing 3D person representations, in real-time systems, often have limitations such as imperfect, incomplete or coarse geometric models, low resolution textures or slow frame-rates. This typically depends on the complexity of the method used to reconstruct in 3D, e.g. stereo-vision, point cloud or visual hull, and the number of cameras used.

In this paper, we present works in progress on multi-camera real-time 3D modeling for tele-immersion and collaborative interactions. Such works build on the experience with the Grimage[1] multi-camera platform, initially focused on mono or multi-user interaction with virtual objects and simulations in a single physical acquisition space. We consider here the generalization of the platform to the case of tele-immersion and collaborative environments spread over multiple sites.

The remainder of the paper is organized as follows. Section 2 presents the multi-camera vision system and the archi-

[*]INRIA / Grenoble Universities, email: firstname.lastname@inrialpes.fr
[†]INRIA / Université Bordeaux 1, email: firstname.lastname@labri.fr

[1] http://www.grimage.inrialpes.fr/

tecture that allow real-time 3D content creation. Section 3 describes how we improved this environment to support multi-site interactions. Section 4 discusses the results obtained through the first experiments, before concluding in section 5.

2. A MULTI-CAMERA VISION SYSTEM

To generate real-time 3D content for tele-immersion and remote collaboration, we have built an acquisition space surrounded by a multi-camera vision system. This section will focus both on the technical characteristics needed to obtain a stream of 3D models and the visualization and interaction capabilities of our system based on these 3D models.

2.1. Acquisition

Cameras. Our acquisition platform is equipped with eight cameras. They are standard firewire cameras acquiring up to 30 fps and 2 Megapixels color images. Each camera is connected to a cluster node dedicated to image processing steps.
Calibration. The multi-camera system needs to be calibrated using an off-the-shelf calibration procedure to estimate each camera's intrinsic and extrinsic parameters. These positions and projection parameters are required for the 3D modeling algorithm.
Synchronization. Data acquisition is synchronously achieved across all cameras by hardware genlocking. Simultaneous acquisitions guarantee that cameras effectively image a geometrically coherent scene content, which can then be reconstructed by video-based modeling algorithms.
Background Subtraction. The physical backgrounds of the acquisition space are assumed static, which allows for efficient segmentation of the region of interest in each image. The object silhouettes thus obtained in each view are used to compute the 3D model of the user, using our shape-from-silhouette reconstruction algorithm. Background subtractions are achieved on each image processing node to maximize CPU and bandwidth efficiency.

2.2. 3D Modeling

Visual Hull. Using the silhouettes extracted from the video streams we compute the *visual hull* of the objects present in the acquisition space. Geometrically, the visual hull is the intersection of the *viewing cones*, the generalized cones whose apex are the cameras' projective centers and whose cross-sections coincide with the scene silhouettes. When considering piecewise-linear image contours for silhouettes, the visual hull becomes a regular polyhedron. We use the EPVH algorithm [2], which computes the complete and exact visual hull polyhedron with regard to silhouette inputs. It provides a full 3D mesh of the objects present in the acquisition space.
Distributive Strategy. To achieve real-time 3D modeling, we developed a parallel version of the exact polyhedral visual hull algorithm. It relies on a three step pipeline, each step being itself parallelized. This approach enables to reach high frequencies (up to 30 frames per second) with a low latency (less than 100ms from acquisition to visualization with 10 dual-Xeon 2.6 GHz).
Texture Extraction. We also extract from each silhouette region the photometric data that will be used later during the rendering process for texturing the 3D mesh.

2.3. Interaction and Visualization

The 3D model and its texture are sent to rendering (Fig. 1). This opens the possibility for live 3D scene content broadcasting, provided we can efficiently transfer 3D models across sites, which we discuss below. The 3D mesh is also sent to the physical simulation component for mechanical interactions.
Simulation. Coupling real-time 3D modeling with a physical simulation enables interaction possibilities that are not symbolic and feel more natural to the user. We have developed Sofa[2], a distributed simulator that handles collisions between soft or rigid virtual objects and the user's body. Unlike most devices used for interactions it allows to use any part of the body or any accessories seen inside the acquisition space without being invasive. Certain interactions are intricate, the prehension of objects for example is very difficult as there is no force information linked to the model.
Rendering. Rendering the 3D model is quite simple as it is already a polygonal surface and texture mapping is done with the extracted textures from the silhouettes. Texture coordinates of 3D model vertices can be efficiently generated using shader-based rendering, by reprojecting vertex coordinates in source images consistently with camera calibration parameters. The exact polyhedral visual hull algorithm guarantees that the 3D model can be projected back to the original silhouette with minimal error, a property that leads to a better quality texture mapping with initial images resolution. Having access to the full 3D surface enables interactive and unconstrained selection of rendering viewpoints, and yields realistic views of the reconstructed person. The rendering can be performed on heterogeneous display devices such as standard monitors, multi-projector walls, head-mounted displays or stereoscopic displays.

2.4. Implementation

FlowVR. The application represents a large code involving various external libraries. It is complex to develop and maintain and requires to execute efficiently on a distributed and parallel environment. To face this software engineering issue we relied on our FlowVR[3] middleware to combine the different software components and distribute them on the nodes of the PC clusters in order to reach real-time executions.

[2] http://www.sofa-framework.org/
[3] http://flowvr.sourceforge.net/

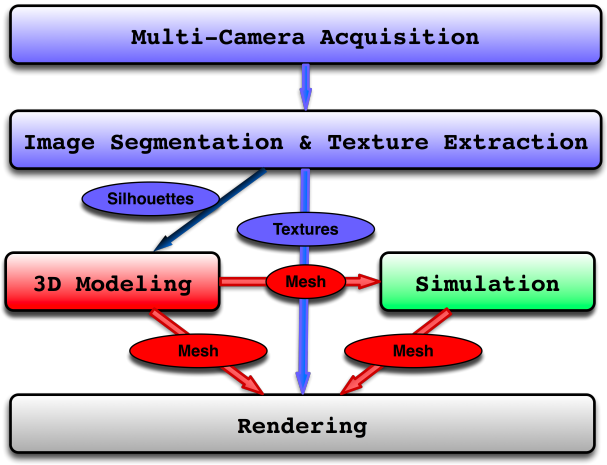

Fig. 1. The application processing pipe-line for a multi-camera acquisition space.

3. A COLLABORATIVE ENVIRONMENT

Tele-immersive Environment. Virtual environments, such as multi-player games or social network worlds, need a representation for each user, often an avatar controlled with keyboard and mouse. Using our system one can be virtualized in real-time and send his life-like representation to the virtual environment. Our contribution is to demonstrate the interest of 3D modeling for such applications through first experiments with two 3D modeling platforms.

Visual Presence. From the user's point of view, the sense of presence is strongly improved as the user avatar is a real representation of himself and not an impersonal 3D model from a database. His actions are not limited to pre-programmed instances. People can recognize themselves and have life-like conversations. It is easier to feel the other users' emotions as it is possible to look directly at their face expression and see their body gesture. Such a configuration really makes them feel as if they were in the same physical space.

Mechanical Presence. Sharing our appearance is not the only interest of our environment, 3D meshes can also be used to interact with shared virtual objects. The server managing the virtual environment receives user information (geometric 3D model, semantic actions…), runs the simulation and sends back the transformation of the virtual scene to the users (Fig.2). The dynamic deformable objects are handled by this server while heavy static scenes can be loaded at initialization on each user's rendering node. Such an environment can be used by multiple users to interact together from different locations with the same virtual objects. For each iterative update the physical simulation detects collisions and computes the effect of each user interaction on the virtual world. It is of course impossible to change the state of the input models themselves as there are no force-feedback devices on our platforms. Physically simulated interactions between participants are also impossible for the same reason.

Data Transfer. Remote site visualization of models requires the transfer of 3D model streams and their associated textures under the constraints of limited bandwidth and minimal latency. The mesh itself is not bandwidth intensive and can be easily broadcasted over the network. The textures (one per camera) induce much larger transfers and represent the bulk of the data load. We will provide data bandwidth measurements in the section 4 for a particular setup. We do not consider any specific transfer protocol, which is beyond the scope of this work.

The FlowVR middleware handles the synchronization of both texture and mesh streams to deliver consistent geometric and photometric data. It also prevents network congestion by resampling the streams (discarding some 3D frames) in order to send only up-to-date data to the end-user nodes.

As the physical simulation only needs the mesh to run, it can run asynchronously, the 3D mesh being sent as fast as possible by each site.

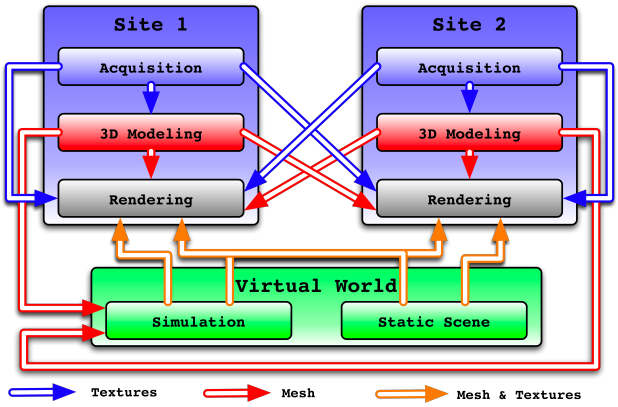

Fig. 2. Application architecture for 2 multi-camera acquisition spaces and a virtual physical environment.

4. EXPERIMENTS

Practical Setup. First experiments are conducted on two platforms located in the same room. We were not able at this time to run experiments with more platforms or distant sites, as we have no access yet to more multi-camera platforms:

- The first acquisition platform is built with 8 firewire cameras with 1MP resolution, allowing an acquisition space of two by two meters, suitable for a full person. The PC cluster used is composed of 10 dual-xeon PCs connected through a dual gigabit Ethernet network.
- The second acquisition platform is a portable version of the first one with an acquisition space of 1 square meter at

table height used for demonstration purpose. It uses 6 firewire cameras and is suitable for hand/arm interactions. The cluster is built with 6 mini-PCs used for camera acquisition, 1 dual-xeon server for computation, and a laptop for supervision.

The 2 platforms are connected by a gigabit Ethernet network using one PC as gateway between the 2 platforms. This PC gathers the data from the 2 platforms and handles the physical simulation.

Data Estimation. Our 8 camera platform produces 1MP images, yielding 3MB images and thus a theoretical 24MB multi-image frame throughput. In practice the only image data needed for texturing lies inside the silhouettes, which we use to reduce transfer sizes. When one user is inside the acquisition space the silhouettes occupy usually less than 20% of the overall image in a full-body setup. Thus an average multi-texture frame takes 4,8 MB. We also need to send the silhouette mask to decode the texture. A multi-silhouette mask frame takes about 1 MB. The overall estimated stream is about 5,8MB. To decrease the needed bandwidth we decided to use the full resolution of the camera for 3D model computation but only half the resolution for texture mapping, reducing the full multi-texture frame to a maximum of 1,45 Mb to transfer at each iteration. The mesh itself represents less than 80 KB (about 10000 triangles).

Running at 20 frames per second, which is reasonable for good interactions, the dual-platform requires a 29 MB/second bandwidth for 3D frame streaming which is easily scalable to a Gigabit Ethernet network (120 MB/s).

Results. We are able to acquire images and to generate the 3D meshes at 20 fps on each platform. The simulation and the rendering processes are running respectively at 50-60 fps and 50-100fps, depending of the load of the system. As they run asynchronously from the 3D model and texture generation we need to resample the mesh and the texture streams independently. In practice the mesh and texture transfer between sites oscillates between 15fps and 20fps, depending on the size of the silhouette inside the images. Meanwhile the transfer between the 3D modeling and the rendering node inside a platform and the transfer going to the simulation node are always running at 20fps. We do not experience any extra connection latency between the two platforms. During execution, the application does not overload the gigabit link.

From the user's point of view the sense of presence is strong (Fig.3). Users do not need to learn the interaction paradigms that are in fact the same as people use to experience the real world. Please refer to the videos[4] of our experiments.

5. CONCLUSION

We presented a first attempt to enable telepresence and collaborative interactions using 2 multi-camera platforms, real-time 3D modeling, and a physical simulation to handle the virtual interaction space.

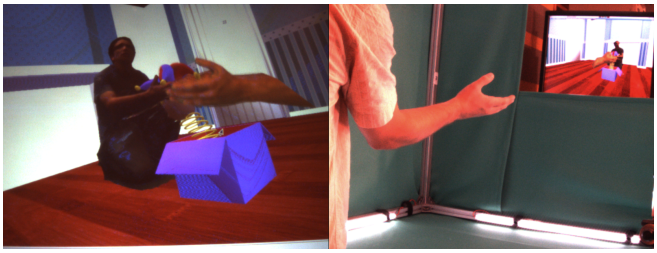

Fig. 3. Left: The 3D virtual environment with a "full-body" user and a "hand" user, interacting together with a virtual puppet. Right: one of the acquisition platforms.

Our first results are very positive in two aspects. They demonstrate the feasibility of such environments and the advantages for the visual and the mechanical presence of the users in a virtual environment. The second aspect is the importance of data transfer in a distributed context, which we will continue to develop in the future. For example, texture sizes could be lowered by using video stream compression, which can be performed directly on each acquisition node while computing the 3D model. Silhouette masks could also be compressed using RLE compression. We could also use the end-user point of view in the virtual scene to select the textures to send to the remote visualization.

Future works will also address long distance experiments using a fast Internet connection and two "full-body" platforms between Grenoble and Bordeaux.

Acknowledgment

This work was partly funded by Agence Nationale de la Recherche, contract ANR-06-MDCA-003.

6. REFERENCES

[1] Kurillo, G., Bajcsy, R., Nahrstedt, K. and Kreylos, O., "Immersive 3D Environment for Remote Collaboration and Training of Physical Activities," in *IEEEVR08*, 2008, pp. 269–270.

[2] J.S. Franco and E. Boyer, "Efficient polyhedral modeling from silhouettes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[4] http://grimage.inrialpes.fr/telepresence/