Augmented Reality for Real and Virtual Humans

Selim Balcisoy, Rémy Torre, Michal Ponder, Pascal Fua and Daniel Thalmann
Computer Graphics Laboratory (LIG), Swiss Federal Institute of Technology
EPFL, CH-1015 Lausanne, Switzerland
Firstname.Lastname@epfl.ch

Abstract

Current Virtual Reality technologies provide many ways to interact with Virtual Humans. Most of those techniques, however are limited to synthetic elements and require cumbersome sensors. We have combined a real-time simulation and rendering platform with a real-time, non-invasive vision based recognition system to investigate interactions in a mixed environment with real and synthetic elements. In this paper we present the resulting system, the example of a checkers game between a real person and an autonomous virtual human to demonstrate its performance.

1. Introduction

Until recently virtual humans have been populating homogenous virtual environments. A wealth of research deals with different aspects of human modeling: Design and Rendering of the 3D model, motion generation, behavioral animation and autonomous virtual humans [3,7]. Virtual Reality systems provide a coherent virtual environment for interactions. Virtual humans can be either autonomous or guided (avatars).

Augmented Reality (AR) can be viewed as a combination of distinct technologies spanning from virtual reality to computer vision [6]. AR enhances the user's view of the real world by adding computer generated information. AR systems acquire relevant information from the real world through sensors; perform a simulation and generate a mixed image combining real and synthetic views. In this paper we present a system that allows virtual and real humans to coexist.

Any AR system with Real and Virtual Humans requires a method of interaction between these two. We are interested in a specific AR context where the user himself is his avatar inside the mixed environment with an autonomous virtual human. Our goal is also to eliminate most, if not all, of the "visible" sensors or targets from our system. Our approach is to track real human's actions by tracking what he is interacting with in the mixed environment. This approach limits our range but frees us from conventional cumbersome sensors. We define objects of interest beforehand and track them during the experiment.

We first present an overview of our AR system. We then introduce the system components: computer vision tools, Virtual Human Director (VHD) and mixing tools. We demonstrate our ideas in the context of a checkers game between Real and Virtual Human.

2. System Overview

Our AR system relies on computing separate tasks on separate workstations to achieve low latency. One workstation is dedicated to computer vision related tasks, another to simulation and rendering of the synthetic image. Computer generated image is mixed with filmed real image by of the shelf video mixer.

There are three main vision related tasks. Calibration of the real camera (performed once before any event starts), tracking of the real camera position and orientation, and tracking of objects of interest. The camera calibration is performed in beforehand and the output is transferred in form of projection matrix to VHD manually. Camera tracking and object sensing is performed in real-time and the output is transferred via Ethernet to relevant VHD modules. (object data for simulation and camera parameters for rendering)

Our second workstation is running an extended version of VHD for AR. It is performing the virtual environment simulation and rendering. Computer generated image is put on the video output.

Figure 1 illustrates major system components and their relationships.

Figure 1: AR system overview

3. System Components

3.1 Vision Tools

In this section we present briefly the vision tools we are using in our AR system. We refer the interested reader to an earlier publication [5] for additional details.

Camera Calibration:

Standard camera calibration software is used [8] to calibrate the camera and compute internal and external camera parameters that define the projection matrix. Figure 2 shows 3D model of a checkers board projected on a real image with correct camera calibration.

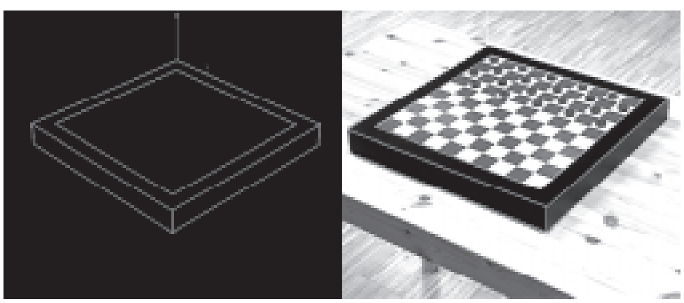

Figure 2: 3D model of checkers board with hidden surface removal.

Camera Ego Motion Tracking:

In a dynamic camera system camera ego motion is to be tracked to preserve correct external camera parameters. We developed a Model Based Optimization approach [5]. It is a paradigm in which an objective function is used to express both geometric and photometric constraints on features of interest. A parametric model of a feature, such as a checkers board, is extracted from one or more images by automatically adjusting the model's state variables until a minimum value of the objective function is obtained.

We acquire a video image, apply a gradient operator, predict the motion based on the previous observation and finally optimize the camera parameters to ensure that the model projects where it should.

Thanks to this tracking algorithm we can now move either the camera or the checkers board during our experiments. The tracker is initialized semi-automatically. Using full frame PAL (720x576) video format on an SGI Octane workstation, we track at a rate of 22.0 Hz. Figure 3 displays three snap shots from a real-time experiment.

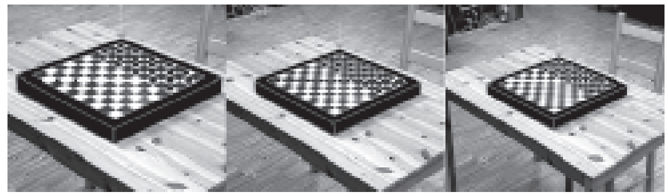

Figure 3: Dynamic camera tracking.

Object Registration and Tracking:

Our camera tracker can be used as an object tracker to track objects of interest. We also have developed a pixel based detection software [5] to detect changes in specific image areas. Using simple object sensors enable us to detect changes in the surrounding environment. In a specific setup like checkers game, tracking checkers is sufficient for virtual human to perceive its real opponent's actions.

3.2 Virtual Human Director

A powerful simulation and rendering platform is essential for AR. In the previous work we developed the Virtual Human Director (VHD)[1] to perform high-end production with Virtual Humans in Virtual Environments. VHD incorporates fully integrated virtual humans with facial and body animation and speech. It provides a range of animation, interaction, scripting and camera control capabilities integrated into one single environment.

VHD is a system based on client-server approach using TCP/IP protocol as a communication layer. The system kernel that is responsible for audio-visual simulation is composed of 5 parallel processes that take control over different aspects of the simulation. Synchronization and communication between the processes listed below is realized through the use of shared memory and semaphore mechanism in those cases when synchronized access to the shared data is required.

Main: process responsible for simulation update which additionally maintains synchronization between all the processes involved

Face: process responsible for generation of the facial and lip animations during the execution of facial expressions and speech

Speech: process responsible for speech generation and its synchronization with lip animations

Communication: this process provides network based communication of the VHD kernel with the client applications used for simulation authoring and interactive control

Now we are going to discuss in more detail the Communication process. Its architecture is based on the big loop that continuously monitors the network for incoming connections (disconnection) requests and messages. The loop hosts 4 VHD servers providing main control mechanism over the whole system through the use of the VHD clients being stand alone applications providing intuitive graphical user interfaces that can be used for authoring and interactive real-time control of the content and course of the simulation:

MAIN Server
This server is used by the MAIN Client for system initialization and interactive real-time control. MAIN Client allows the user for choosing the simulation environment, actors and actors' voices. After system initialization it allows for interactive control over the facial expressions, speech, keyframes, walking, grasping, real-time motion capture animation using Magnetic Sensor Flock Of Birds technology. It provides as well tools for camera control and recording of the simulation in the form of the script that can be then loaded and replayed. Finally this server is used to terminate the simulation and execution of the VHD system shutdown. An example control interface, Actor Controller is shown on Figure 4. The left hand of the interface is used to control facial expressions and speech, the right hand for body animation. Special features like motion capture, walking engine and motion blending are placed on the right side also. There are two additional interfaces for camera control and scripting similar to Actor interface.

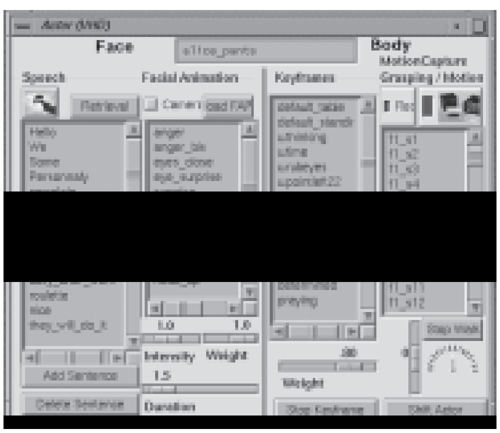

Figure 4: Actor Control Interface.

NAV Server
This server can be used optionally which means that the respective NAV Client application can be used as an optional virtual authoring and interactive control tool in all those cases when more precise actor navigation capabilities are required. By optional we mean that the client application can be connected and disconnected in any time depending on actual needs of the user. NAV Client allows for interactive navigation and design of navigation paths using symbolic top-view navigation map. It allows as well for definition of interest or gaze points that are attracting actor visual attention during the navigation depending on actor current position, direction, importance of the gaze points, etc. Figure 5 presents a snap shot from NAV client application during a demonstration:

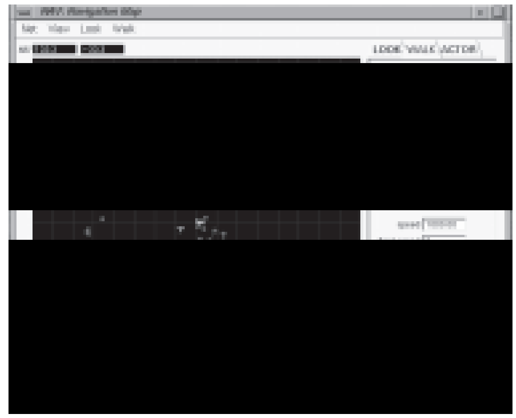

Figure 5: Navigation Client

WP Server
This is another VHD server that can be used optionally by the respective WP Client application. Its purpose is to provide precise control of the actor walking style through the manipulation of 40 independent parameters bound to amplitudes and offsets of joints taking parts in animation. WP Client allows for design of walking styles and then for blending them depending on the value of inherent and user-defined parameters like linear velocity, angular velocity, actor sadness, actor tiredness, etc.

EXTRA Server
This is an additional optional server that duplicates some of the functionality of the servers listed above. At the same time it provides some extra control mechanisms particularly important in the filed of Augmented Reality. The server allows for real time control of all camera parameters. The control can be realized both by sending in the camera position, orientation and filed of view values or by sending in the complete 4x4 double precision transformation matrices. The latter solution assures higher precision and allows to work with transformation matrices obtained directly from the real camera calibration and vision based real-time tracking being executed on the remote machine. It is important to not here that the camera control capability of the EXTRA Server is the crucial feature in case of the Augmented Reality applications. It allows for separation of the two most computationally heavy processes: VHD kernel and real camera position tracking system. In this way the real camera tracking system can be executed on the dedicated machine able to process video input. An effect of tracking only small number of bytes containing transformation matrices is required to be transmitted to the VHD kernel for further processing. Additionally since the frame rate of the visual simulation is targeted at the 25fps there is no need to communicate the transformation matrices more frequently.

In effect the communication does not impose any high bandwidth requirements on the network.
All communication between the VHD servers and client applications is based on the special communication library that provides higher level functionality over the TCP/IP layer like automatic broken connection detection, connection authentication, message validation, automatic resynchronization in case of message corruption, etc. The library provides as well an easy way for communication protocol definition and its further modification if necessary. All communication protocols used by servers and client applications are designed in the way that only minimal number of bytes is required to be communicated through the network. In order to avoid network congestion only integer ID's and the most essential parameters of the actions to be performer by the actors are transmitted. Since the whole simulation is being executed on the VHD kernel side there is no need for synchronization of client applications. In effect query-ack message loops are avoided which eliminates the delays caused otherwise by the round trip times.

3.3 Mixing

To mix video sequences from graphics engine and filmed scene, we use conventional chroma-keying technique. In chroma-keying technique an intermediate colour, acts as a "keyhole" and determines where a video layer shows through another. Compositing is performed by Sony DFS 300P mixer. We set blue as keying colour. Figure 6a shows the virtual scene with virtual human. Some real objects of interest, chair and table are presented in virtual environment with 3D models in keying colour. This ensures correct clipping of real and virtual elements. Figure 6b is a snap shot from live output from video mixer.

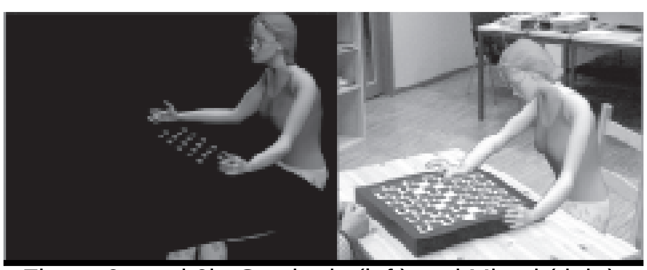

Figure 6a and 6b: Synthetic (left) and Mixed (right) Images.

4. A Case study: Checkers Game

To test our system for a realistic interactive task, we designed a checkers game scenario. Playing board games against expert systems has been around more than two decades. We used a public domain checkers simulator to control a virtual human. Instead of text based or graphical interface the real player has now a realistic looking opponent in front of him with real checkers and board. We used VHD to control body animation, facial expressions and lip synchronized speech. We generated facial expressions and sentences for the virtual human before the demonstration. All the body animation for the game playing is controlled by the checkers game driver and the animation is generated by an inverse kinematics engine [9] inside VHD. Figure 7 shows sequential snap shots from game.

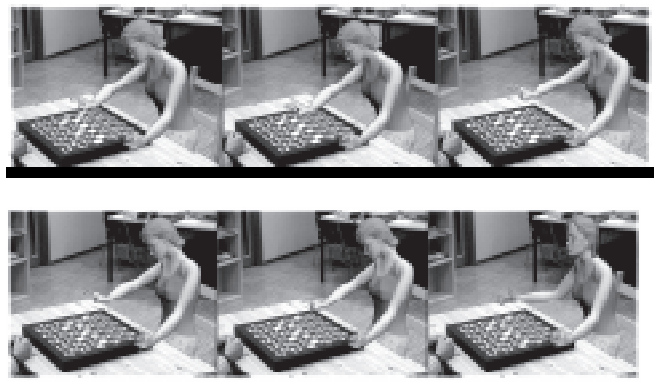

Figure 7: Snap shots from Checkers Game with camera movement.

5. Discussion and Conclusion

In this paper we have presented a novel approach to Real and Virtual Human Interaction in an AR context. A non-invasive vision based system is combined with a real-time virtual human simulation and rendering platform. Our model based approach does not require cumbersome sensors or specific targets to solve the registration problem and acquire information from the real world.
We have demonstrated our approach in the context of a checkers game. The tracker was running with 22 Hz on a double processor SGI Octane with a digital video option. VHD was running with 24 Hz on a SGI Onyx2 workstation with six processors. The Figure 8 is an enlarged snap shot from our experiments, showing the current visual performance of our AR system.

Figure 8: Zoomed snap shot.

As future work we would like to use our AR system to investigate complex interactions. We plan to extend our vision system to enhance virtual human's perception of its surroundings.

6. References

[1] G. Sannier, S. Balcisoy, N. Magnenat-Thalmann, D. Thalmann, VHD: A System for Directing Real-Time Virtual Actors, The Visual Computer, Springer, Vol.15, No 7/8, 1999, pp.320-329

[2] S.Balcisoy, D.Thalmann, Interaction between Real and Virtual Humans in Augmented Reality, Proc. Computer Animation'97, IEEE CS Press, 1997, pp.31-38.

[3] R. Boulic, T. Capin, Z. Huang, L. Moccozet, T. Molet, P. Kalra, B. Lintermann, N. Magnenat-Thalmann, I. Pandzic, K. Saar, A. Schmitt, J. Shen, D. Thalmann, The HUMANOID Environment for Interactive Animation of Multiple Deformable Human Characters, Proc. Eurographics '95, Maastricht, August 1995, pp.337-348

[4] P. Fua. RADIUS: Image Understanding for Intelligence Imagery, chapter Model-Based Optimization: An Approach to Fast, Accurate, and Consistent Site Modeling from Imagery. Morgan Kaufmann, 1997. O. Firschein and T.M. Strat, Eds.

[5] R. Torre, P. Fua, S.Balcisoy, M. Ponder, D. Thalmann, Interaction Between Real and Virtual Humans: Playing Checkers, EGVE 2000, Amsterdam June 2000.

[6] R.T. Azuma. A survey of Augmented Reality. Presence, Teleoperators and Virtual Environments, 6(4):335-385, August 1997.

[7] T. Capin, I. Pandzic, N. Magnenat-Thalmann, D. Thalmann, Avatars in Networked Virtual Environments, John Wiley and Sons, 1999.

[8] J.P. Tarel, J.M Vezien, Camcal Manual: A Complete Software Solution for Camera Calibartion, Technical Report 0196, INRIA, September 1996.

[9] R. Boulic, Mas , Hierarchical Kinematics Behaviors for Complex Articulated Figures in: Interactive Computer Animation, Prentice Hall, 1996, pp.40-70.