

An Immersive Game - Augsburg Cityrun

Klaus Dorfmueller-Ulhaas, Dennis Erdmann, Oliver Gerl, Nicolas Schulz,
Volker Wiendl, and Elisabeth André

Augsburg University, Eichleitnerstr. 30, 86159 Augsburg, Germany
dorfmueller-ulhaas@informatik.uni-augsburg.de
<http://mm-werkstatt.informatik.uni-augsburg.de>

Abstract. We present a platform for creating immersive 3D games including a new interface for navigating through virtual scenes. One innovative part of this application is a crowd simulation with an emergent behaviour of virtual characters. While the users has to move very quickly through a crowd of nearly two hundred virtual characters he is supported by a precise, fast operating, and unobtrusive navigation interface.

1 Introduction

A novel interaction technology is shown by Augsburg cityrun. While the user moves in front of a 3D projection screen, he is wearing shutter-glasses enabling a 3D impression (compare Fig. 1). Virtual characters autonomously move through



Fig. 1. *Left:* The user controls his navigation by bending his body. Wireless tracking is performed with infrared cameras to enable an unobtrusive interface. *Right:* Crowd simulation is done by enabling emergent behaviour of each character.

the inner city of Augsburg. The user's task in this game is to catch a specific pedestrian, but to avoid hitting others in a crowd within a close time limit. The player controls the continuation of his journey only by moving his upper body. An optical tracking system recognizes markers attached to the player's shutter-glasses and is thus able to capture the player's movements. The impression of immersion is achieved by combining the optical tracking system with the navigation control and a special 3D projection system (see Fig. 2) that adapts



Fig. 2. *Left:* The shuttered stereo projection system. *Right:* The virtual model of the exhibit at the CeBit 2006 in Hannover.

itself dynamically to the viewpoint of the user in front of the projection screen. This feeling is enhanced by 3D sound effects. The complete gaming system is distributed among two independent computers. The first computer tracks the user's position by optical tracking. The data of the the player's eye positions is sent via UDP sockets to the game system. The receiver is responsible for doing the entire game processing, including sound and stereoscopic graphics rendering.

2 Crowd Simulation

In order to make the game more vivid and to avoid a blank and lifeless city, we populated the scene with lifelike and convincingly acting characters. For this reason a crowd simulation module has been implemented. The agents in this system are represented as particles similar to Reynolds famous boids [Reynolds, 1987]. Collision avoidance between characters is achieved by calculating physically based repulsion forces between particles, which are accumulated to a steering force determining the proximate position. [Heigeas et al., 2003] propose a three zone model for setting the strength of the repulsion forces. The resulting force function can be adjusted by the physical parameters stiffness and viscosity. We have adopted the model, but calculate these parameters automatically. In the process a continuous function has been created providing smoother movements of our agents. The realtime rendering of multiple animated characters is still a serious challenge. Many crowd visualization systems are using some sort of impostors [Aubel et al., 2000], [Dobbyn et al., 2005] to reduce the polygon count. Impostors are prerendered images that show a character with important animation keyframes from different camera angles. Usually they have a flat appearance, are very static, and can consume much texture memory. Since we are low on memory due to the large amount of texture data in the city scene we opted for a real 3D representation of characters in combination with skeletal animation. The skinning is done on the GPU with vertex shaders in order to get the best performance and keep the CPU free for other tasks. Another promising optimization that fits perfectly for a city scene is occlusion culling [Sekulic, 2004]. This GPU-accelerated technique helped us to determine which

characters are hidden behind buildings and hence don't have to be transmitted to the rendering pipeline.

3 User Navigation and Perspective Correction

Fast navigation through a crowd requires a very sensitive and accurate as well as unobtrusive and intuitive input system. There are systems using no markers for user tracking such as a system described in [Wren et al., 1997]. However, our application is highly dynamic and thus requires a navigation method that cannot be warranted by pure natural feature trackers. We therefore employ an optical stereoscopic tracking system that works similar to the approach described in [Dorfmueller, 1999]. The interaction between the virtual player's movement in the virtual world and the real user's position is done by a simple but efficient approach. We define a coordinate system in the real user's interaction area. The origin of this coordinate system is defined during the external camera calibration process. In case, the user stands close to the origin no movements in the virtual space are produced. Otherwise, the position of the user is converted into a polar coordinate system. The outcome of this procedure is an angle to one of the coordinate axes and the distance to the origin as shown in Fig. 3. The angle is used for the direction of movements and the distance controls velocity. In case of a desktop VR system that generally uses a small display, it is assumed that the user is observing the scene close to an axis perpendicular to the center of the monitor screen. However, this assumption does not hold when viewing virtual objects on huge screens e.g. in CAVEs [Cruz-Neira et al., 1993]. Within our application the user is not restricted in his movements. In fact, he is motivated to perform large movements in front of the screen as depicted in Fig. 3. Therefore, a perspective correction is necessary and has been done by implementing an asymmetric viewing frustum. Furthermore, the scene becomes alive through three-dimensional sound effects emitted in relation to the user's position together with an asymmetric viewing perspective.

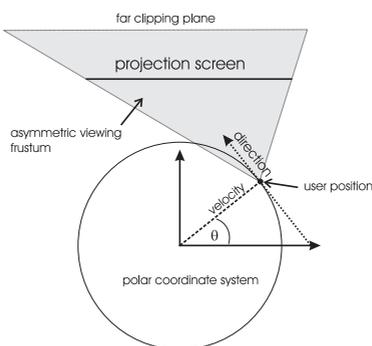


Fig. 3. User position given in polar coordinates controls navigation. The user position requires a perspective correction with an asymmetric viewing frustum.

4 Concluding Remarks

We described a system that consists of a 3D sound system, a 3D projection screen with shutter-glasses and an optical tracking system that transmits positional changes to the 3D world. Due to the optical tracking system, no cables are required to capture user movements, but only the markers attached to the glasses. We presented the system at the Open Lab event where more than one hundred people interacted with it. Even though the users had no experience with 3D tracker interfaces they were able to intuitively interact with the game without feeling hindered by obtrusive cables. The funny body poses of the players during the navigation and the witty comments of the virtual passers-by additionally contributed to the joy of playing. We hope to collect more user data when the complete system is demonstrated at CeBit 2006 in Hannover (see Fig. 2). The interface control in combination with real-time visualization and animation of virtual characters is so far unknown in game applications. Through the skilful combination of an optical tracking system with the navigation control and the projection system, a unique 3D experience is achieved.

References

- [Aubel et al., 2000] Aubel, A., Boulic, R., and Thalmann, D. (2000). Real-time display of virtual humans: Levels of detail and impostors.
- [Cruz-Neira et al., 1993] Cruz-Neira, C., Sardin, D., and Defanti, T. (1993). Surround-screen projection-based virtual reality: The design and implementation of the cave. In *SIGGRAPH 93 conference proceedings*, pages 135–142, Onahium. ACM Press.
- [Dobbyn et al., 2005] Dobbyn, S., Hamill, J., O’Conor, K., and O’Sullivan, C. (2005). Geopostors: a real-time geometry / impostor crowd rendering system. In *SI3D ’05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 95–102, New York, NY, USA. ACM Press.
- [Dorfmueller, 1999] Dorfmueller, K. (1999). An optical tracking system for vr/ar-applications. In Gervautz, M., Hildebrand, A., and Schmalstieg, D., editors, *Virtual Environment’99*, pages 33–42. Springer Verlag, Wien, New York.
- [Heigeas et al., 2003] Heigeas, L., Luciani, A., Thollot, J., and Castagné, N. (2003). A physically-based particle model of emergent crowd behaviors. In *Graphicon*.
- [Reynolds, 1987] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH ’87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA. ACM Press.
- [Sekulic, 2004] Sekulic, D. (2004). Efficient occlusion culling. In Fernando, R., editor, *Gpu Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Addison Wesley Pub Co Inc.
- [Wren et al., 1997] Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.