RTFX: On-Set Previs with UnrealEngine3

Lesley Northam, Joe Istead, and Craig S. Kaplan

University of Waterloo,
200 University Ave. Waterloo, Ontario, Canada
{lanortha, jwistead, csk}@uwaterloo.ca

Abstract. This paper discusses on-set previsualization with distributed motion capture, virtual camera and asset control, and real-time rendering using a video game engine. Our test harness, RTFX, demonstrates the feasibility and usefulness of a system that couples Epic Games' UnrealEngine3 with the Houdini 3D animation kit by Side Effects Software and a passive motion capture system by Vicon.

Keywords: previsualization, real-time rendering, virtual production.

1 Introduction

Previsualization (previs) tools employ 3D modeling and animation software to prototype film sequences before production begins. These tools allow creative and technical issues to be explored early in the production pipeline, improving film quality while reducing costs and human error [1], [2]. *On-set* previs and virtual production tools augment the camera feed with rudimentary special effects and virtual sets, providing real-time feedback to the production cast and crew [3], [4]. These tools have been integrated with motion capture (mocap) and green screening to create elaborate virtual worlds and characters for several productions such as *Avatar* [5].

Using the UnrealEngine3 video game engine, we have developed an on-set previs tool called RTFX that provides real-time visualization of mocap, virtual cameras and 3D effects (including particle systems). These features allow actors to interact with virtual assets and trigger special effects in the virtual set. The tool also combines distributed mocap and effects streams into a single visualization. These features enable us to demonstrate the viability of several novel applications:

1. **Detailed special effects and virtual asset previs with integrated mocap.** This improves the quality of feedback to the director concerning actor-asset relationships in the virtual set, e.g., when an actor triggers a startling explosion special effect, do they turn their head in the correct direction?
2. **Distributed mocap.** Traditional systems require all the actors to work in the same volume at the same time, so these systems are subject to scheduling conflicts and volume limitations, e.g., *Beowulf*, the largest volume on record, limited the virtual set to 21 actors [6]. RTFX connects to multiple mocap servers, each with their own recording volume, then combines the data from each server in real-time to generate a common virtual set.

3. **Distributed visualization.** RTFX supports previs from several configurable virtual cameras, enabling the director to explore the virtual set with a variety of camera parameters, including those calibrated to actual equipment. Also, mapping these cameras to actor headsets ensures coherent action across multiple mocap volumes.
4. **Distributed skeletal control.** RTFX supports distributed control of individual skeletons. This enables large, sophisticated meshes to be animated by a mixture of sources (both mocap and programmatic). For example, the mocap bones from two humans could control a dragon's wings and body, while an external program coordinates the motion of the dragon's head with fire-breathing special effects.

2 Previous Work

Several video game engines have been used to control virtual assets and render real-time previs. Industrial Light & Magic used an Unreal engine to plan camera sequences for the movie *Artificial Intelligence* [7]: the mouse, keyboard and joystick movements that normally control video game character motion were harnessed to control camera sequences within the virtual set.

Nitsche's [8] experiments with camera controlling interfaces and puppetry also use an Unreal engine. This system captures hand-controlled puppet-based motion and projects it into the virtual set. Mazalek [9] proposes a similar system called TUI3D (Tangible User Interface for real-time 3D) that controls animated characters through real-time puppet interaction. TUI3D acquires joint position and orientation data from puppet sensors, and then passes this data into a custom Unreal2004 tool that provides enhanced scripting and game controls for animation. Also, Mazalek et al. extended TUI3D with a wearable interface for real-time avatar control [10].

Nitsche [8] and Mazalek [9] also identify a potential limitation for video game engine-based real-time previs. To optimize in-game frame rates and detail levels, these engines provide limited animation control and normally favour pre-fabricated sequences over live animation. But since movie action tends to be non-deterministic and non-repetitive, animating these sequences requires a resource-intensive approach that cannot take advantage of the underlying engine optimizations.

3 Observations

We demonstrate the practicality of a game engine-rendered previs system by measuring frame rate under different workloads, and then justify the results from the context of the RTFX architecture (Figure 1). Note that frame rates below 24 fps (the minimum film frame rate) may produce undesirable flickering and hesitation.

The first test measured the relationship between mocap complexity (actor count and distribution) and frame rate. In particular, we recorded frame rate during visualizations of one through twenty-one mocap actors. The experiment was repeated with three distributions of actors: local (all actors in one mocap volume), remote (all actors in a remote volume), and distributed (half local actors, half remote).

Fig. 1. RTFX architecture

Figure 2 shows that frame rate decreases as the number of mocap actors increases, and visual performance degrades with more than seven local, four distributed or three remote actors (frame rate falls below 24 fps). The architectural cause of this trend is the RTFX backend: for every frame, and for every bone in the mocap volume, a serial call to the mocap server is executed. Parallelizing the calls improves the curves.

Fig. 2. Relationship between mocap complexity and frame rate for three actor distributions: local, remote and distributed

The second test measured the effect of virtual set (game map) detail level on frame rate while visualizing mocap. Two detail levels of virtual sets were used: basic (untextured floor plane and light) and complex (Epic's FoliageMap: a jungle scene with particle systems, lighting effects and hundreds of animated assets). We observed

that set detail level has only a small effect on performance compared to the number of actors: on average the complex set's frame rate was only 2 fps lower than that of the basic set.

Finally, we tested the effect of virtual cameras on performance. RTFX supports three camera types: static (fixed position), motion capture, and Houdini. The frame rate of each camera was recorded within basic and complex virtual sets. No observable difference was recorded: all experiments ran at 62 fps. This result was expected since camera simulation requires much less network communication than mocap visualization.

Acknowledgements. RTFX is a collaborative research project with the Screen Industry Research and Training Centre (SIRT). We'd like to thank them for their continued assistance with the development and testing of RTFX.

References

1. The Previsualization Society: What is Previs?,
<http://www.previsociety.com/what-is-previs/>
2. Dickreuter, R.: Interview with Colin Green (2007),
<http://www.xsibase.com/articles.php?detail=133>
3. Bennett, G., Denton, A.: Developing practical models for teaching motion capture. In: Proceedings of SIGGRAPH Asia, pp. 1–5. ACM, New York (2009)
4. Green, C.: Previsualization for "Starship Troopers" managing complexity in motion control. In: Proceedings of SIGGRAPH, p. 294. ACM, New York (1998)
5. Momcilovic, D.: Virtual Production Stage. In: Proceedings of SIGGRAPH. ACM, New York (2010)
6. Motion Capture World Records,
<http://www.motioncapturesociety.com/resources/world-records/>
7. Lehane, S.: Unrealcity. ILM Creates Artificial Cities for Artificial Intelligence. In: Film and Video, vol. 18(7) (2001)
8. Nitsche, M.: Experiments in the Use of Game Technology for Pre-Visualization. In: Proceedings of Futureplay, pp. 160–166. ACM, New York (2008)
9. Mazalek, A., Nitsche, M.: Tangible interfaces for real-time 3D virtual environments. In: Proceedings of the International Conference on Advances in Computer Entertainment Technology, pp. 155–162. ACM, New York (2007)
10. Mazalek, A., Chandrasekharan, S., Nitsche, M., Welsh, T., Clifton, P., Quitmeyer, A., Peer, F., Kirschner, F., Athreya, D.: I'm in the game: embodied puppet interface improves avatar control. In: Proceedings of the 5th International Conference on Tangible, Embedded and Embodied Interaction, pp. 129–136. ACM, New York (2011)