Augmented Reality Techniques in Games

Dieter Schmalstieg
Graz University of Technology, Austria

Fig. 1: Comic chat in *There!*

Fig. 2: Follow-me car in *Colin MacRae Rally 4*

Fig. 3: Remote view as inset in *XIII*

1   Introduction

As a consequence of technical difficulties such as unreliable tracking, many AR applications get stuck in the "how to implement" phase rather than progressing to the "what to show" phase driven by information visualization needs rather than basic technology. In contrast, most of today's computer games are set in a fairly realistic 3D environment, and unlike many AR applications, game interfaces undergo extensive usability testing. This creates the interesting situation that games can be perfect simulators of AR applications, because they are able to show perfectly registered "simulated AR" overlays on top of a real-time environment. This work examines how some visualization and interaction techniques used in games can be useful for real AR applications.

2   Game interfaces and controls

Games are constrained by the hardware typically owned by casual users: a stationary monoscopic screen, keyboard, mouse and gamepad. Stereoscopic displays are not available, but the game world is usually shown with a wide field of view. This situation can be compared to a video see-through head mounted display (HMD) with a wide-angle camera, but not with the narrow field of view of an optical see-through HMD. Moreover, commercial games do not support 3D tracking devices, which means all interaction examples found in games will be "at a distance" and use image plane techniques [6], as users must rely on tools such as ray-picking to select and manipulate objects. For these reasons, game interfaces will probably be most relevant to mobile video see-through AR setups dealing with objects at a distance.

One caveat is that head-tracked AR normally implies a continuous camera control model (i. e., the user cannot freeze camera movement entirely), while games use explicit camera control (for example, rotating the camera with the mouse while controlling other aspects of the game with the keyboard). Explicit camera control may be advantageous in some situations, where high accuracy operations are required (unwanted degrees of freedom can be effectively locked), while continuous camera control enables a higher degree of spontaneity. Nevertheless, many game control techniques are still applicable to AR.

3   Selection

Selection of objects in games is mostly done by "point and click" with the mouse, which translates into ray-picking [2] in an AR environment. To overcome the deficiencies of raypicking of small or occluded objects, and also for selecting groups of objects, games often rely on rubberbanding techniques, mostly with viewport-aligned rectangles. Some games use rectangles projected onto the ground in world coordinates, but this method suffers from the fact that there is no intuitive choice of the rectangle axes, since world-aligned axes appear arbitrarily oriented in camera space. In AR applications with lots of objects, rubberbanding may be an interesting extension to selection.

First-person shooter games use a cross-hair for targeting and selection. The cross-hair is coupled to the camera movements. Head-tracked AR allows a similar approach, while keeping the hands free. However, a small field of view and the dynamics of physical movement may make it more difficult to accurately target an object. Therefore, another technique used as a relief in difficult games could be useful in AR, namely automatic snapping

to the object closest to the cursor, if such an object is sufficiently close. This technique is frequently combined with filtering to select only specific objects. Many game interfaces also use button-free selection after hovering briefly over an object. This should also be suitable for AR selection, especially if small movements due to tracker jittering are ignored.

4   Highlighting and Annotation

One of the main purposes of AR systems is directing a user's attention to specific context-dependent information. Highlighting of objects, persons, or other parts of the environment is therefore a fundamental technique. The primary objective of a highlighting style is maximizing attention while minimizing occlusion or other visual deterioration of the scene. In games, a popular technique involves rendering highlighted objects with increased brightness, because this technique requires only a small rendering modification in texture attenuation, and does not alter visibility or overall style of the visual scene. In video see-through AR, a tracked physical object for which the silhouette or a 3D model is known could be highlighted by attenuating the background video buffer directly.

Another technique used in many games to minimize visual clutter while displaying information in context is to place icons and glyphs in a fixed relative position to the object. For example, an established convention is the annotation of standing characters with a ground-aligned, often color-coded circle, loosely resembling a ground shadow. For airborne objects, such a convention can be extended with vertical anchor lines to convey height above the ground. In addition to ground shadows for characters, the ground is often used as a 2D canvas to display additional information, such as routing or directional information, or destination areas for certain operations. Utilizing the ground rather than using floating annotations may be an interesting design alternative for AR visualizations.

If floating annotations, such as character names or chat bubbles (Fig. 1) and displayed above the characters' heads, are used in games, they often lead to display clutter, which is sometimes resolved by prioritizing annotations. It remains an open question whether more sophisticated filtering and label layout techniques such as presented in [1, 3] are not implemented because of the increased complexity or because they are not preferred by game players.

Another form of annotation involves symbolic objects added to the scene, often for navigation purposes, such as waypoint landmarks. Movable objects, such as "follow-me" cars (Fig. 2) and characters also make for useful interface elements which could be used in more complex AR applications.

5   Non-registered overlays

Game displays are typically divided between a 3D view and a 2D heads-up display for menus, inventories, 2D maps and status reports. Games which require a user to manipulate a large number of virtual objects often allow direct drag and drop between the 3D view and the 2D

view. Information-heavy AR applications could benefit from such direction manipulation techniques during information gathering. A related idea concerning combined 2D/3D interfaces extends annotations of 3D objects by linking and referencing complex 2D views and menus with callout lines similar to technical diagrams.

6   Visualization modes

AR displays have the freedom of completely altering a user's perception of the environment. For example, games use X-ray or thermal vision to reveal hidden or occluded information. Apart from cutaway views, which are also popular for AR systems [5], games often use color-coded silhouettes of occluded objects, which is less invasive to the rendering of the game world.

Another popular visualization method involves multi-view techniques, such as magnification inlays or the possibility of viewing a remote viewpoint, for example, of a secondary character. Some AR researchers are already experimenting with using live imagery from remote cameras to present such remote views [4]. However, if a 3D model of the environment is available, an AR system could use a secondary view to present a navigable purely virtual view of a remote viewpoint (Fig. 3). Such a view could incorporate any kind of live information (for example, current position of a remote tracked person) and would not be constrained by the availability of remote cameras.

7   Conclusions and future work

Games provide a large variety of tried and tested techniques for efficient interface techniques that can be useful for AR. By studying and using games as AR simulators, researchers can build on this knowledge for designing more compelling AR applications. Future work will involve the implementation of a toolkit composed of those techniques.

8   References

1. Bell B., S. Feiner, and T. Höllerer. View management for virtual and augmented reality. In Proc. UIST'01, pp. 101–110, Orlando, Florida, USA, November 11–14 2001. ACM.
2. Bowman D., E. Kruiff, J. LaViola, I. Poupyrev: 3D User Interfaces: Theory and Practice. Addison-Wesley 2004.
3. Julier S., M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, and T. Höllerer. Information filtering for mobile augmented reality. In Proc. ISAR 2000, pp. 3–11, Munich, Germany, October 5–6 2000.
4. Kameda Y., T. Takemasa, Y. Ohta: Outdoor See-Through Vision Utilizing Surveillance Cameras. Proc. ISMAR 2004, p. 151- 160, 2004.
5. Livingston M., E. Swan, J. Gabbard, T. Höllerer, D. Hix, S. Julier, Y. Baillot, D. Brown: Resolving Multiple Occluded Layers in Augmented Reality. Proceedings ISMAR 03, Tokyo, Japan, 2003.
6. Pierce J., A. Forsberg, M. Conway, S. Hong, R. Zeleznik, M. Mine: Image plane interaction techniques in 3D immersive environments. Proceedings of the 1997 symposium on Interactive 3D graphics, pp. 39-47. 1997.

*Web page with more examples from games:*http://www.icg.tu-graz.ac.at/Members/schmalstieg/homepage/aringames/