A realistic Augmented Reality Racing Game using a Depth-Sensing Camera

Adrian Clark
The HIT Lab NZ
University of Canterbury, Christchurch, New Zealand
adrian.clark@hitlabnz.org

Thammathip Piumsomboon
The HIT Lab NZ
University of Canterbury, Christchurch, New Zealand
thammathip.piumsomboon@pg.canterbury.ac.nz

Abstract

As augmented reality (AR) applications become more common, users are expecting increasingly sophisticated experiences combining impressive visuals, interaction, and awareness of the environment. Existing technology capable of understanding user interaction and the environment is often expensive or restrictive. However the newly released Microsoft Kinect provides researchers with a low cost and widely available real time depth sensor. In this paper, we investigate using the Kinect as a means to give AR applications an understanding of the three-dimensional (3D) environment they are operating in, and support new ways for the user to interact with virtual content in a natural and intuitive way.

CR Categories: H.5.1 [Information Interfaces and Representation]: Artificial, augmented, and virtual realities;

Keywords: Interactive AR, acquisition of 3D environment, entertainment, Microsoft Kinect

1. Introduction

With the increasing processing power and popularity of mobile computing devices such as smart phones, tablets and handheld game consoles, AR is becoming more accessible to the public, and even entry level computers are capable of rendering complex real time AR scenes. While current consumer technology is capable of displaying AR scenes, interaction with virtual content in those scenes is often performed using restrictive interaction metaphors such as the paddle [Hornecker 2009], or expensive specialist hardware such as a tablet and a stylus [Szalavari 1997].

To improve the AR experience, virtual content should behave realistically in the physical environment it's placed in. Fiducial marker and natural feature registration algorithms are able to calculate the pose of a given target, but have no awareness about the environment the target exists in. This lack of awareness can cause the virtual content to float above real objects or appear inside them, or occlude objects it should appear behind, breaking the illusion that the virtual content exists in the real world.

In this paper we explore using the Microsoft Kinect [Leyvand 2011] as a low cost consumer device to enable environmental awareness and natural interaction with virtual content in AR. Previous research has demonstrated using specialized hardware for providing depth data [Iddan 2001] [Garcia 2010]. The Kinect is used to examine a 3D volume within the task space, and with the transformation between the Kinect and the AR viewing camera known, virtual content can be realistically composited in the environment. The user can then interact with the content in a natural and intuitive way using their hands or real objects.

The main contributions of our work are:

- Integrating the Kinect in an AR framework to provide 3D information about the environment.
- Development of an environmentally aware AR application to create a more realistic AR experience.

2. Related Work

Awareness of the environment within AR is a well researched area. It is required to achieve correct occlusion [Lepetit 2000], collision detection [Breen 1995], and realistic illumination and shadowing effects [Wang 2003]. While these features are not necessary for AR, it has been shown that applications which include such cues can establish a stronger connection between real and virtual content [Sugano 2003].

Early attempts at environment awareness required manual modeling of all the real objects in the environment, and online localization of the camera to ensure virtual objects interacted with real objects appropriately [Breen 1995][MacIntyre 2005]. This method is both time consuming and inflexible, as any changes in the environment will require recalibration.

Later approaches involved more automatic techniques, such as contour based object segmentation [Berger 1997], depth information from stereo cameras [Zhu 2010] and time-of-flight cameras [Fischer 2007], or online SLAM [Ventura 2009]. By automatically acquiring the relevant information from the scene there are no offline calibration requirements and the system can correctly process the environment even when objects change or are added and removed.

3. Integration of Kinect with AR framework

To create an interaction volume, the Kinect is positioned above the desired interaction space facing downwards, as shown in Figure 1. A reference marker is placed in the interaction space to calculate the transform between the Kinect co-ordinate system and the co-ordinate system used by the AR camera.

The OPIRA natural feature registration library [Clark 2008] using SURF features [Bay 2008] is used for all natural feature based registration due to the robustness to perspective distortion and other image transformations of OPIRA, and the fast computation time of SURF.

In the initialization phase, the transformation from the marker to

[1] http://www.openni.org/ retrieved on 26/05/11

VRCAI 2011, Hong Kong, China, December 11 – 12, 2011.
© 2011 ACM 978-1-4503-1060-4/11/0012 $10.00

the Kinect is calculated using natural feature based registration. With the transformation known, the corners of the marker are projected into the two-dimensional (2D) Kinect colour and depth images, and the 3D position for each corner is calculated using the OpenNI framework[1].

The size of the marker is calculated in millimeters using the Euclidian distance between corners, and the AR co-ordinate system is established at this scale with the origin at the top left corner of the marker. Finally, the transformation between the corner positions in the Kinect co-ordinate system and the corner positions in the AR co-ordinate system is calculated and stored.

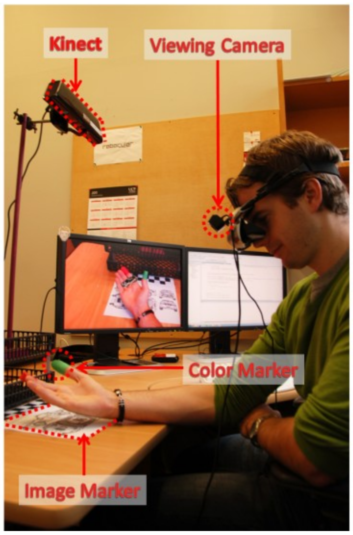

Figure 1. *The interaction set up. The Kinect is suspended above the interaction volume, with the reference image marker below it.*

With the transformation between the Kinect and the AR co-ordinate systems known, 3D data from the Kinect can be transformed into the AR co-ordinate system. Figure 2 shows the point cloud data captured by the Kinect projected into the AR co-ordinate system in mesh form.

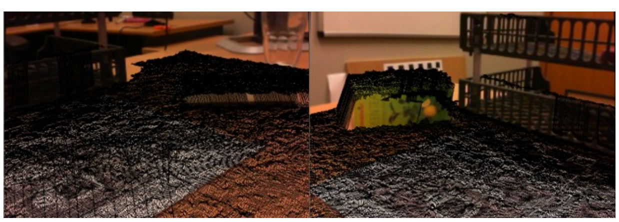

Figure 2. *Kinect depth data displayed as mesh in the AR scene.*

Assuming the plane that the marker lays on is the ground plane, object segmentation can be easily achieved by using a simple threshold of the distance of each pixel from the ground plane.

4. AR Micromachines

As an example of environmental awareness, an AR application was designed and developed to allow users to control small virtual cars in the interaction space. A physics model allows the car to react to real-world objects in real-time using depth information provided by the Kinect, allowing users to create obstacle courses using real objects.

As the Kinect is capable of capturing 3D data in real time, in addition to adding obstacles such as walls and ramps, the user is able to lift the car in their hand or on an object. Figure 3 shows examples of the AR Micromachines application.

Figure 3. *AR Micromachines car driving application. The car is able to drive up ramps (top right), sit on top of objects (bottom left), and even be lifted off the ground (bottom right).*

4.1 Terrain Construction

The depth image obtained by the Kinect is prone to missing values due to shadowing of the infrared data. To resolve this, missing values are identified and an inpainting algorithm is used to estimate their values. The depth image is then resized from 640x480 resolution to 160x120 resolution, and the coordinate system aligned such that the upper left corner of the image-based marker represents the origin in both the real and virtual world.

The 160x120 image is converted into a tri-mesh where each pixel's value represents the height of the pixel from the ground plane. The tri-mesh is used by the physics engine for collision detection. Optionally, the tri-mesh can be converted into a mesh using Delaunay triangulation and rendered in AR to show the physics representation of the world, as shown in Figure 4.

4.2 Implementation of Physics Model

The Bullet physics library[2] was used to provide a realistic physics simulation. Bullet is an open source physics engine which provides collision detection and rigid and soft body dynamics. For these examples only rigid body dynamics were used.
The car is represented in the physics world using the Bullet Raycast Vehicle object, which provides physical properties such as tire friction, engine and breaking forces, and suspension characteristics to increase realism.

As the tri-mesh in the physics simulation is being updated in real-time, the user can interact with the virtual content with realistic physics. For example, the user's hand or a book is viewed as part of the dynamic scene, and can be used to pick up or push the car around. However, this interaction is limited, especially for more complex shaped objects, due to the single point of view of the Kinect. More realistic interaction would require multiple views or 3D tracking of objects to determine the orientation and complete shape of the object.

[2] http://bulletphysics.org/wordpress/ retrieved on 26/05/11

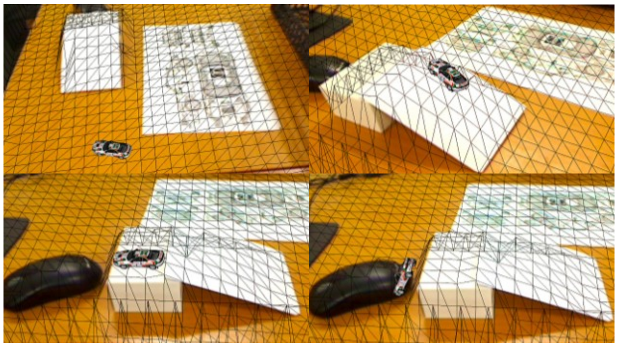

Figure 4. *The Micromachines AR application. A wireframe mesh representing the physical height map is overlaid on the scene. The car accelerates up the paper ramp (top right), approaches the edge of the platform (bottom left), and due to the force of gravity falls off the platform (bottom right).*

4.3 Rendering

The OpenSceneGraph [Burns 2004] framework was used for rendering in the application. The input video image is rendered as the background, with all the virtual objects rendered on top. The virtual objects are transformed so as to appear attached to the real world.

The terrain data is rendered as an array of quads, with an alpha value of zero. This allows realistic occlusion effects of the terrain and virtual objects, while not affecting the users view of the real environment, as shown in Figure 5 (Top).

As planes with an alpha value of zero cannot have shadows rendered on them, a custom shader was written which allowed shadows from the virtual objects to be cast on the invisible terrain map. The shadows add an additional level of realism, as well as important depth cues which allow users to immediately understand the distance that the car is from the ground, which is particularly useful when the car is performing jumps or falling. The shadow casting can be seen in Figure 5 (Bottom).

Figure 5. *Rendering effects. Occlusion of virtual objects by real objects (Top). Virtual shadows on real objects (Bottom)*

5. Performance

The goal of this research is to create a more realistic and engaging AR experience. To achieve this, the application must be capable of running in real time, while ensuring the virtual content behaves appropriately in the environment which it is in, and that the user can interact with it in a natural and intuitive manner.

On an Intel 2.4Ghz quad core desktop computer, the environmentally aware example application is capable of running at over 25 frames per second. The Kinect provides an error of less than a centimeter at the ground plane when placed approximately 70 centimeters from the ground plane. This error is small enough that the virtual car appears to realistically interact with real objects.

6. User Experience

Over 50 participants have experienced the AR Micromachines application, ranging in age from elementary school children to adults in their 70s. Of these participants, their experience with AR ranged from minimal to expert users and developers. While there have been no formal evaluation or interviews with the participants at this stage, a number of interesting observations were made.

6.1 Preliminary Observational Study

The AR Micromachines preliminary study prototype allows two users to control two virtual cars using Xbox 360 controllers. An RGB camera is used for the viewpoint, and a projector is used for display, as shown in Figure 6. Participants were given general description of the system and shown how to use the controller. There were no game rules imposed on the participants and they are free to explore the application. Objects such as books and desktop stationary were provided for the users to construct their own obstacle course, and users were allowed to introduce new objects to the scene at any time.

With only minimal explanation of the system, participants were able to engage in the game quickly. Some groups designed their own tracks and challenged their friends to races. Others examined the interaction and explored the limitation of the system by directly manipulating the car with objects and hands.

6.2 Observational Findings

All users were impressed with the realistic physics which were possible with the environment awareness afforded by the Kinect depth information. Even novice users of AR were able to intuitively understand how the interaction area could be changed by placing real objects in the view of the Kinect. Many participants said they found the game play unique and fun, and several expressed an interest in purchasing the game if it were commercialized, even despite its simple game play mechanics.

One of the most interesting observations made was the high level of engagement of users which often resulted in even those who were extremely familiar with AR to forget the restrictions of the technology. Even though it was explained that the marker had to remain at least partly in view for the system to know the position of the viewing camera, users would often follow their car with the camera, forgetting to keep the marker in view or blocking the marker view with their arms.

Many participants expected to interact with the car as if it was a real toy car. For example some participants attempted to grasp the car with their hands in order to move it around the table. Others tried to flatten the car with a book. Both actions resulted in the car falling through the physical terrain and disappearing from the table.

Figure 6. *The AR view is projected on to the wall enables a shared display (Top). Participants use Xbox controller to control the virtual cars and modify the obstacle course (Bottom).*

7. Discussion and Conclusion

In this work we investigate using the Microsoft Kinect as a means to create AR experiences with environmental awareness and natural methods of interaction. Previously such systems have been cumbersome, expensive or inaccessible to consumers. In contrast the Kinect provides a cheap and accessible research platform.

A tabletop driving game was created to explore environmental awareness and interaction methods with virtual content. Depth information about the environment was obtained using the Kinect, and an integrated physics engine was used to provide a more realistic AR experience. Real-time updates of the environmental data allowed users to interact their virtual car with real objects in the scene, or even with their hands.

8. Future Work

In future, we wish to further examine how AR experiences can be made more realistic using the Kinect. We plan to investigate additional methods of object segmentation to provide more realistic interactions between the real and virtual worlds through hand gestures. We also want to implement multiple viewing camera support such that each user can have their own view of the environment.

Eventually, we plan to evaluate these approaches to improving the AR experience to quantify how effective these techniques are.

References

HORNECKER, E., DUNSER, A. 2009; Of pages and paddles: Children's expectations and mistaken interactions with physical-digital tools, Interacting with Computers, Volume 21, Issues 1-2, Special issue: Enactive Interfaces, January 2009, Pages 95-107

SZALAVARI, Z., GERVAUTZ, M. 1997; "The Personal Interaction Panel - a Two-Handed Interface for Augmented Reality", Computer Graphics Forum, 1997, pages 335-346

IDDAN, G. J., YAHAV G. 2001; "Three-dimensional imaging in the studio and elsewhere", Three-Dimensional Image Capture and Applications IV, Vol. 4298, No. 1. (2001), pages 48-55.

GARCIA, J., SHPUNT, A. 2010; "Depth ranging with moire patterns", U.S. Patent Application 20100201811, December 2010

LEYVAND, T., MEEKHOF, C., WEI, Y., SUN, J. AND GUO, B. 2011,"Kinect Identity: Technology and Experience," Computer, vol. 44, pages 94-6, 2011.

LEPETIT, V.; BERGER, M.-O. 2000; "Handling occlusion in augmented reality systems: a semi-automatic method," IEEE and ACM International Symposium on Augmented Reality, 2000. (ISAR 2000). Proceedings, pages 137-146, 2000

BREEN D E, ROSE E, WHITAKER R T. 1995 "Interactive Occlusion and Collision of Real and Virtual Objects in Augmented Reality" Technical Report ECRC-95-02, ECRC, Munich, Germany, 1995.

WANG, Y., SAMARAS, D. 2003; "Estimation of multiple directional light sources for synthesis of augmented reality images". Graph. Models Volume 65, Issue 4, Pages 185-205, July 2003

SUGANO, N., KATO, H., TACHIBANA, K. 2003; "The effects of shadow representation of virtual objects in augmented reality", Proceedings International Symposium On Mixed And Augmented Reality (ISMAR), pages: 76-83, 2003

MACINTYRE, B., GANDY, M., DOW, S., BOLTER, J.D. 2005; "DART: a toolkit for rapid design exploration of augmented reality experiences", Proceedings of ACM Trans. Graph, 2005

BERGER, M.O. 1997; "Resolving occlusion in augmented reality: a contour based approach without 3D reconstruction," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 91-96, 17-19 Jun 1997

ZHU, J., PAN, Z., SUN, C., CHEN, W. 2010; "Handling occlusions in video-based augmented reality using depth information". Computer Animation and Virtual Worlds, Volume 21, Issue 5, pages 509-521, September 2010

FISCHER, J., HUHLE, B., SCHILLING, A. 2007; "Using Time-of-Flight Range Data for Occlusion Handling in Augmented Reality", Eurographics Symposium on Virtual Environments (EGVE), pages 109-116, 2007

VENTURA, J., HOLLERER, T. 2009; "Online environment model estimation for augmented reality". In Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '09), pages 103-106, 2009.

CLARK, A., GREEN, R. AND GRANT, R.: 2008, "Perspective correction for improved visual registration using natural features", Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference, pages 1-6.

BAY, H.; ESS, A.; TUYTELAARS, T.; VAN GOOL, L. 2008 "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pages 346-359, 2008

BURNS, D., OSFIELD, R. 2004; "Open Scene Graph A: Introduction, B: Examples and Applications," Virtual Reality Conference, IEEE, page 265, IEEE Virtual Reality Conference 2004 (VR 2004), 2004