

Achieving Object Persistence In An Augmented Reality Indoor Environment

Manuel Daniel Gutiérrez, Aida Margarita Márquez, Helmuth Trefftz
EAFIT University, Virtual Reality Laboratory
{mgutiere, amarque4, htrefftz} @eafit.edu.co

ABSTRACT

In this paper we present a system for achieving object persistence in Augmented Reality Indoors Environments. In AR systems based on the Augmented Reality toolkit without object persistence, virtual objects appear and disappear as the corresponding pattern gets in and out of the user's field of view. Object persistence allows the environment to remember the relative position and orientation of virtual objects, making them appear more stable to the user. We describe object persistence as a series of transformations applied to the objects in a simple indoor environment. We then present the results of a quantitative experiment that shows how interactivity is improved by the use of object persistency.

KEYWORDS

Augmented Reality, Visual Tracking, Object Persistence.

1 Introduction

Augmented Reality applications based on the ARToolkit [1] make use of patterns as means to determine the position and orientation of objects with respect to the camera. A pattern is an image (Figure 4) or object that has certain contents that allow ARToolkit to calculate its relative pose with respect to the camera.

Most applications make use of small objects that can be placed over a given pattern. Virtual objects are drawn if and only if the corresponding pattern is in the user's field of view. This might not be convenient, for instance, when the object is large, because the pattern might easily get out of sight when looking at some parts of the object. In solving this problem, we propose to provide two types of patterns in the environment: patterns of the first type are used to compute the position and orientation of the user in the environment, and those of the second type are used to determine the position of the persistent virtual objects. The relative position of the patterns must be known. This allows the system to compute the position of a virtual object even if the corresponding pattern is not inside the user's field of view. When the relationship between the patterns has been established, they become a "visual tracker", allowing the application to compute the view point transformation and the position of the persistent virtual objects at any time by having as few as one pattern inside the field of view. With the research

reported in this paper we want to prove that users can achieve better results in the tasks they are performing in an augmented reality environment when the system *provides persistency*, i.e. objects remain in place even if one of the patterns is not inside the user's field of view.

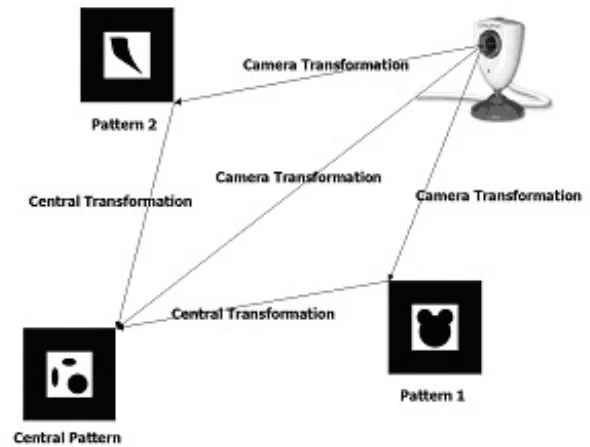


Figure 1. Pattern Relationship.

2 Related Work

In order to provide immersion in the virtual world it is important to have a direct relation between the movement of the body, or the head and the effects of those movements in the virtual environment. This is used to control the vision of the digital surroundings, the displacement of the viewpoint and the location and movement of different entities in the virtual world. This capability is fundamental to provide a sense of presence in the virtual environment [2].

The recognition of the user's movements is obtained with trackers located in some parts of the body (head), which can use different systems (ultrasonic, magnetic, mechanical or optical, or even hybrid systems). Trackers must provide real time figures of three-dimensional positions (X, Y, Z distances) and directions (yaw, pitch and roll), which are known as "six degrees of freedom".

The simplest are gyroscopes, positioned typically in the posterior part of the HMD to recognize the turns of the head and to adapt the image of the screens according to the natural rotation of the view, granting the sensation to be surrounded by the virtual model [3]. Correct tracking

allows the VR system to correctly describe the virtual surroundings to the user "relative" to the location in the virtual world [4]. Trackers, that identify the absolute space of interaction between the physical atmosphere and the virtual model (also known as "working envelope") [1], [5].

A study made in the University of Southern Australia describes an outdoor application in first person of ARQuake [6], [7], [8], [9], that make use of a DGPS for user positioning and a TCM2-80 sensor for user orientation and provides interaction with the user in an outdoor application. An outdoor application is a program that runs in an open environment [10], [11]. Our solution work for indoor applications, and is based on the AR toolkit.

Park et. al [12], present two different methods to increase robustness and accuracy in real-time AR tracking.

3 Experimental Investigation

In the experiment, users were presented with a big airplane (Figure 2) that was rotated 90 degrees around its horizontal axis. One wheel was subtracted from the airplane's landing gear. Subjects were placed in the middle of the environment and asked to look for any missing part of the plane. Times to complete the task were recorded for each subject. The experiment was conducted with twenty subjects. Ten subjects navigated the environment using an AR environment providing persistency. Times to complete the task ranged between 10 and 41 seconds. Ten subjects used an AR environment without persistency. Times to complete the task ranged between 21 and 124 seconds in this case. The first set of users commented on the convenience of knowing always where the object was. The second set of users used the system without persistency. They learned to keep the pattern inside the field of view in order to see the virtual object. The second set was then presented with the persistency system. They said that they could know where the object was and where the anomaly was more easily and the task was easier to achieve. The users of the non-persistence system also said that it was very hard when the object disappeared and then they needed to look at the pattern again to regain the object visibility.

three-dimensional position, require a reference fixes and have a limited reach. These way trackers have a limited



Figure 2. Virtual Airplane. This object was downloaded from <http://www.3dcafe.com/asp/freestuff.asp>

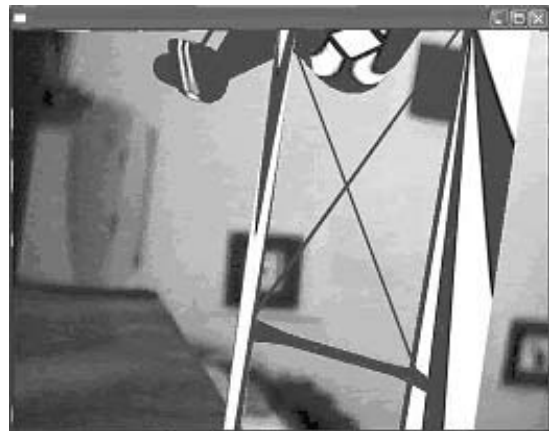


Figure 3. System with object persistence. This figure shows a screen shoot of one user's point of view with Object Persistence when the central pattern is not visible.

4 Solution

The problem was then solved relating each pattern with a special pattern that we use to represent the center of our environment. We used two flags, one would tell us if the pattern had already been seen, which meant that the pattern's central transformation had been set. The other flag would tell us if that pattern had been totally configured which meant that that pattern had a central transformation that had been proved to be valid.

The process involved in configuring a pattern follows:

```

Begin with the first element of the list.
While "end of the list of patterns not reached"
Do
  If "the pattern is visible" Then
    If "the pattern has been configured"
      then
        Set that pattern as the main
        pattern_

```

```

End If.
Else If "the pattern has not been
configured"
Then
    Set its transformation between the
    camera and the pattern.
    Set its visible flag to true.
End If.
End While.

Begin with the first element of the list.
While "end of the list of patterns not reached"
Do
    If "The pattern is visible" Then
         $B_{center} = A_{center} + (A_{camera} - B_{camera})$ 
        Pattern configured status = true.
    End If.
End While.

```

The center pattern is a pattern that has its transformation configured to zero before execution, the resultant value represents the transformation between the pattern we are observing and the center pattern (Figure 1).

This is the way a pattern is configured:

$$B_{center} = A_{center} + (A_{camera} - B_{camera})$$

Where

- A_{camera} is the vector of the *configured pattern* with respect to the camera,
- B_{camera} is the vector of the *pattern to be configured* with respect to the camera,
- A_{center} is the vector of the *configured pattern* with respect to the center,
- B_{center} is the vector of the *pattern to be configured* with respect to the center,

When the process is completed the viewpoint's value is changed to be the addition of the main patterns' transformation relative to the camera and the main patterns transform' relative to the center.

Translation Vector = (X Component, Y Component, Z Component)

Rotation Vector = (α , β , ρ), the normal rotation angles around the axis.

With the computed viewpoint, the environment can be navigated as if it was a real environment because the world is referenced by the center. This is the way we obtain the viewpoint:

$$Vp = A_{center} + A_{camera}$$

Where Vp is the Viewpoint.

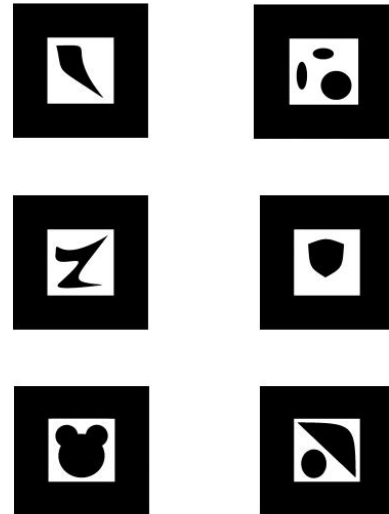


Figure 4. Position Patterns.

4 Hardware and Software utilized

The system was developed using C/C++ as the main programming language, the ARToolkit for pattern recognition and translation into transformation matrices of the position of the patterns related to the camera, OpenGL for virtual object rendering, GLUT for window management, GLAUX for bitmap texturing, a 3DS loader found in www.gametutorials.com, a web-cam for video capture, and Virtual i-o eye glasses for immersion.

5 Conclusions

The experiment's statistical analysis consisted in a Paired Sample Comparison with data obtained. That was the most appropriate technique to use, because each kind of data, The System With Object's Persistence and The System Without Object's Persistence follow a Normal Distribution closely.

This analysis displays the results of three tests concerning the center of the population from which the sample of The System With Object's Persistence - The System Without Object's Persistence comes. The first test is a t-test of the null hypothesis that the mean The System With Object's Persistence - The System Without Object's Persistence equals 0,0 versus the alternative hypothesis that the mean The System With Object's Persistence - The System Without Object's Persistence is not equal to 0,0. Since the P-value for this test is less than 0,05, we can reject the null hypothesis at the 95,0% confidence level. The second test is a sign test of the null hypothesis that the median The System With Object's Persistence - The System Without Object's Persistence equals 0,0 versus the alternative hypothesis that the median The System With Object's Persistence - The System Without Object's Persistence is not equal to 0,0. It is based on counting the number of values above and

below the hypothesized median. Since the P-value for this test is less than 0,05, we can reject the null hypothesis at the 95,0% confidence level (Figure 5). The third test is a signed rank test of the null hypothesis that the median The System With Object's Persistence - The System Without Object's Persistence equals 0,0 versus the alternative hypothesis that the median The System With Object's Persistence - The System Without Object's Persistence is not equal to 0,0. It is based on comparing the average ranks of values above and below the hypothesized median. Since the P-value for this test is less than 0,05, we can reject the null hypothesis at the 95,0% confidence level.

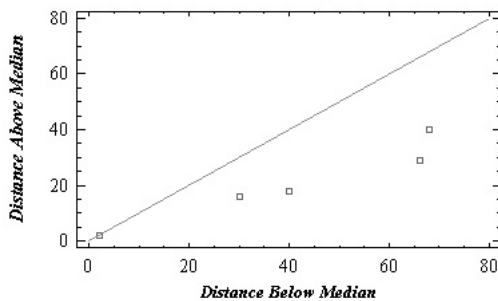


Figure 5. Symmetry Plot. This graphic calculates and displays a plot that is helpful in determining the symmetry of the data. Symmetrical data would closely follow the reference line. In the experiment the data is asymmetric, i.e. Median 1 – Median 2 ≠ cero.

Finally, the principal conclusion of this experiment is that the System With Object's Persistence demonstrated to satisfy better the user's expectation, than The System Without Object's Persistence, while providing also for better interactivity. User's comments also supported the fact that persistency is convenient in order to provide for better interactivity and immersion in an AR environment (Figure 3).

	System With Object's Persistence	System Without Object's Persistence
Count	10	10
Average	22.6	62.7
Variance	84.4889	1145.12
Std. Deviation	9.19178	33.8397
Minimum	10.0	21.0
Maximum	41.0	120.0
Range	31.0	99.0
Std. Skewness	0.859606	0.890537
Std. Kurtosis	0.288242	-0.339661

Table 1. This table shows summary statistics for each of

the selected data variables. It includes measures of central tendency, measures of variability, and measures of shape. Of particular interest here are the standardized skewness and standardized kurtosis, which can be used to determine whether the sample comes from a normal distribution. Values of these statistics outside the range of -2 to +2 indicate significant departures from normality, which would tend to invalidate many of the statistical procedures normally applied to this data.

6 Acknowledgments

We thank the worthy collaboration, uninterested orientation and timely advice, of Carlos David Correa, Maria Cecilia Velez y Martha Alvarez, without which this paper wouldn't have been possible to achieve.

7 References

- [1] B. Thomas, N. Krul, B. Close, and W. Piekarski. Usability and Playability Issues for ARQuake. *4th Int'l Symposium on Wearable Computer*. Pp 139-146, Atlanta, GA, USA, October 2000.
- [2] B.H. Thomas. Using Augmented Reality to Support Collaboration in an Outdoor Environment. *Special Session Augmented Reality: Usability and Collaborative Work in the HCI International 2001, 9th International Conference on Human-Computer Interaction*. Pp 743-747. New Orleans, LA, August 2001.
- [3] T. Höllerer, D. Hallaway, N. Tinna, and S. Feiner. Steps Toward Accommodating Variable Position Tracking Accuracy In A Mobile Augmented Reality System. *2nd International Workshop on Artificial Intelligence in Mobile Systems (AIMS '01)*, pages 31-37, 2001.
- [4] P. Milgram and D. Drascic. Perceptual Effects in Aligning Virtual and Real Objects in Augmented Reality Displays. *41st Annual Meeting of Human Factors and Ergonomics Society, Albuquerque, New Mexico, September 1997*.
- [5] W. Piekarski, and B. Thomas. Augmented Reality With Wearable Computers Running Linux. *School of Computer and Information Science. University of South Australia. Mawson Lakes, SA, Australia*.
- [6] A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingstong. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. *Proceedings of SIGGRAPH 96 (New Orleans, LA, August 4-9, 1996)*. In *Computer Graphics Proceedings, Annual Conference Series 1996, ACM SIGGRAPH*. Pp 429-438.

[7] M. Hou, P. Milgram. Effects of Surface Characteristics on Alignment of Graphics and Real Objects in a Stereoscopic Augmented Reality Environment. *Proceedings of The EIA2000/ HFES2000 Congress*.

[8] B. Thomas, B. Close, J. Donoghue, J. Squires, P.D. Bondi, M. Morris, and W. Piekarski. ARQuake: An Outdoor/Indoor Augmented Reality First Person Application. *In ISWC2000 – 4th International Symposium on Wearable Computers*, 2000, Atlanta, Ga, USA.

[9] W. Piekarski, and B. Thomas. Tinmith – Hand: Unified User Interface Technology for Mobile Outdoor Augmented Reality and Indoor Augmented Reality. *IEEE VR 2002 – Virtual Reality Conference. March 2002. Orlando, FL, USA*.

[10] W. Piekarski, B. Gunther, and B. Thomas. Integrating Virtual and Augmented Realities in an Outdoor Application. *2nd IEEE and ACM International Workshop on Augmented Reality. October 20 - 21, 1999. San Francisco, California*.

[11] H. Kato, and M. Billinghurst. Pattern Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. *Proceedings for The 2nd IEEE and ACM International Workshop on Augmented Reality 1999. Pp. 85-94. October 1999. San Francisco*.

[12] J. Park, B. Jiang, and U. Neumann. Vision – based Pose Computation: Robust and Accurate Augmented Reality Tracking. *2nd IEEE and ACM International Workshop on Augmented Reality. October 20 - 21, 1999 San Francisco, California*.