Hand-Based Interaction in Augmented Reality

Chris McDonald
School of Computer Science
Carleton University
Ottawa, ON, Canada
cmcdona3@scs.carleton.ca

Shahzad Malik
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada
Shahzad.Malik@utoronto.ca

Gerhard Roth
Computational Video Group
National Research Council
Ottawa, ON, Canada
Gerhard.Roth@nrc.ca

Abstract

Interaction with virtual objects in an augmented environment enhances the user's interpretation of their presence. An augmented reality (AR) system that uses computer vision for registration can use the same technology for simple gesture recognition. This paper describes hand detection and simple gesture recognition techniques useful in pattern-based AR systems. Pattern-based registration is briefly discussed followed by the details of hand extraction and analysis. The paper concludes with an application of these techniques using windows-based programming.

1. Introduction

Unlike virtual reality, which encompasses a user in a completely computer-generated environment, in augmented reality (AR) the user sees both the physical world and the virtual augmentation. The virtual objects can consist of text, 2D images, or 3D models. One of the most promising vision-based augmentation techniques involves tracking a planar pattern in real-time and then augmenting virtual objects on top of the pattern [9, 10]. Other tracking techniques exist such as colour blob finding [11] and magnetic or inertial trackers [12, 13, 14]. A further enhancement to such an augmented environment is the ability to interact with the virtual objects as one would with the real ones. Other systems exist, which require special gloves [1], fixed camera orientation [8], or more restrictive pattern visibility [2]. In this paper, we show how to perform real-time hand detection over the planar pattern, and how to use this as the basis for interaction with the augmentation.

2. Pattern tracking system

In plane based AR a black and white planar pattern is tracked through the video sequence, shown in figure 1 on the left. The tracking system computes a planar homograhy [3, 4] between the detected and tracked corners [15] of a known planar pattern. This homography defines a warp from the original two-dimensional pattern to its location in the image frame. From this transformation, the necessary camera parameters can be extracted [5, 6, 7] to define a 3D coordinate system on the image-space pattern. This coordinate system is used as an origin for augmenting the virtual objects, shown in figure 1 on the right. As the target and/or camera move, the homography is updated in real-time. This ensures proper alignment of virtual objects with the real scene in each frame of video.

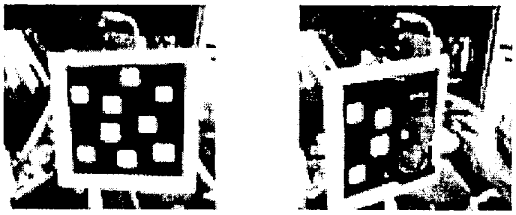

Figure 1. Augmented object on a planar pattern

3. Hand detection

In order to have human interaction with an augmented environment, human gestures must be recognized. Although human gesture comes in many forms, we chose a simple hand gesture as the means for interaction. The hand is detected using image subtraction, at which point detailed information is extracted for gesture recognition.

0-7803-7635-8/02/$17.00 © 2002 IEEE

3.1. Image subtraction using the homography

Image subtraction is a commonly used technique for detecting changes in a video scene. This approach relies on a fixed camera position relative to the reference scene in order to detect the pixel variation due to foreground changes. However, in vision-based AR the camera can move, which normally eliminates image subtraction as a useful tool. When performing augmentation onto planar patterns the pattern space representation of each frame has a fixed position and orientation regardless of camera or target motion. This is because the pattern-to-frame space homography is updated for each frame. When the frame is warped using the inverse homography, the position and orientation of the camera and target are lost. Essentially the camera motion has been removed by this inverse warping. Pattern-space image subtraction can then be performed on this stabilized image, therefore effectively performing image subtraction even for a moving camera.

To subtract the hand image for the stabilized pattern image we must be able to distinguish the hand from the background. Although the black and white pattern colours differ greatly from the colour of the occluding hand, changes in lighting can cause this distinction to be minimal in the captured frame of video. As the lighting conditions change, the intensity of the white regions varies more than the black. For this reason, the black regions are analyzed separately from the white. In both cases, the histogram of the region shows a peak in the pattern colour and a peak representing the hand colour. A threshold is chosen in the valley between the two peaks leaving a binary image. These two binary images are combined to form the binary representation of the subtraction result.

Using the binary image from the thresholding process, the fingers are extracted by a flood-fill algorithm for blob detection. All blobs that meet a minimum pixel count of sixty are stored and assumed to be fingers of the occluding hand.

3.2. Improving the augmentation

With this hand detection mechanism in place, improvements to the visual and functional aspects of the augmentation system can be made.

The standard procedure used by this system to augment a video sequence with virtual objects in real-time is to render the virtual objects over each captured frame. Since the hand is a part of the captured frame, it is thus occluded by any virtual objects. The immersive illusion created by augmenting a video scene is therefore diminished by such occlusion inaccuracies.

Using the point-set representation of the hand, the convex hull of each blob set is computed in order to create a clockwise contour of each hand component. This representation of the hand lends itself to the standard polygon drawing facilities of OpenGL. During a render cycle each polygon, defined by the convex hull of a hand region, is rendered to the stencil buffer. When the virtual object is rendered, a stencil test is performed to omit pixels that overlap the polygons in the stencil buffer. This gives the illusion that the hand is properly occluding the augmentation. Figure 2 shows the augmentation with and without the stencil test, right and left respectively.

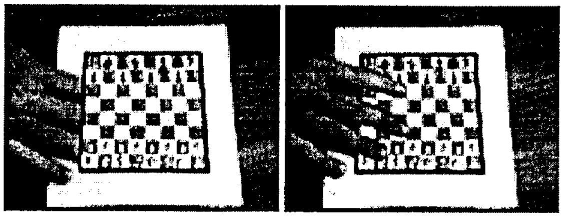

Figure 2. Occlusion using the stencil buffer

Another aspect of the augmentation system that can be improved with this occlusion information is the reliability of the corner tracking. When the hand occludes a corner's search box, sometimes a phantom or false corner is produced. Using our convex blob representation a quick collision scan can be performed to test the containment of any hand pixels in the search boxes. Then we simply invalidate those search boxes that contain hand pixels, shown as dark squares in figure 3. The light squares in figure 3 represent the un-occluded search boxes.

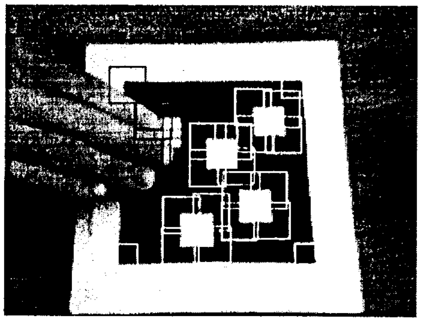

Figure 3. Search box invalidation

This means that those occluded corners will be ignored during the tracking phase, which increases the robustness of the tracking.

4. Hand gesture recognition

To further test our ideas we have implemented the ability to recognize the point and select gesture on the target plane. The information gathered by the hand detection phase greatly simplifies the gesture recognition process. The finger tip location is calculated for the pointing action, and the finger blob count is used for the select action.

4.1. Fingertip location

To determine the location of the user's point and select actions a pointer location must be chosen from the hand point set. To simplify this process, the current system constraints were exploited and a number of assumptions were made. The first useful constraint deals with the amount of target occlusion. The planar tracking system used for augmentation assumes that approximately half of the target corners are always visible. To satisfy this constraint, only a portion of a hand can occlude the planar target at any time. This allows us to make a legitimate assumption that only fingers shall occlude the target. From this we get:

Assumption 1: Separated fingers will be detected as separate blobs in the detection phase.

Due to the simplicity of the desired interaction, a second assumption was made:

Assumption 2: Fingers will remain extended and relatively parallel to each other.

This is also a reasonable assumption because pointing with one or more extended fingers is a natural human gesture. The third constraint used to simplify the process was the following:

Assumption 3: Any hand pixel set will contain at least one pixel on the border of the pattern-space representation of the current frame.

Using all three assumptions the hand gesture recognition process begins by selecting the largest detected finger blob. From the blob's point set, the orientation of the principal axis is calculated using central moments. The principal axis line, shown as the long line cutting the finger blob in figure 4, is defined by forcing it through the blob centroid. The next step involves finding the root point on the principal axis. This represents approximately where the finger joins the hand. This simplification holds as a result of assumption two. Using assumption three, a border pixel, r_b, is chosen from the blob and its closest principal axis point, r_p, is chosen as the root. The finger tip, t_b, is then computed as the farthest point in the blob from the root point, and is used as the pointer location.

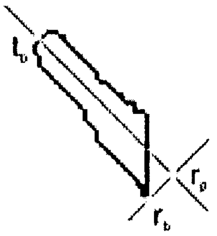

Figure 4. Finger tip location using orientation

4.2. Simple gesture capture

The number of detected finger blobs is the basis for the selection process, using the first assumption of section 4.1. A single detected finger blob represents the gesture of pointing, shown on the left in figure 5. Multiple detected finger blobs represent the gesture of selecting, shown on the right in figure 5. The finger tip location is indicated by the small cross on the prominent finger.

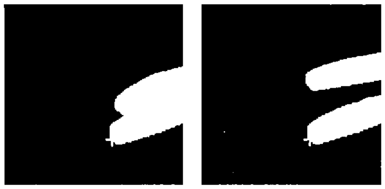

Figure 5. Finger count for gesture recognition

Interaction with the augmentation can be performed, for example, by showing one finger for pointing and introducing a second finger over the target for selecting.

5. Hand interaction on the plane

Using the hand detection and gesture recognition information gathered by the system, a mechanism for two-dimensional interaction has been defined. This type of interaction is similar to the current window-based operating systems that have two-dimensional mouse input. For this reason, a fully-functional control panel dialog box was built to demonstrate the augmented interaction.

5.1. Displaying the interface

Since the hand gestures are captured over the planar pattern, the control panel must be augmented onto the pattern within the AR system. This means that the bitmap representation of the control panel dialog box, shown on the left in figure 6, needs to be stored as a graphics texture and then rendered over the pattern as a virtual object, shown on the right in figure 6.

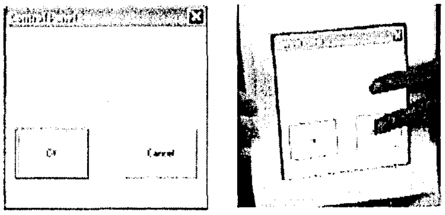

Figure 6. Control panel dialog box

This representation is updated regularly to continuously capture the state of the control panel interface. This mechanism provides the user with the ability to see the visual behaviour of the interface in real-time.

5.2. Interaction with the interface

With the visual feedback mechanism in place, a method is required to initiate the interaction in order to orchestrate system behaviour.

The fundamental interaction parameters are the gesture type and the location of interest. In this system, the first time that multiple fingers are detected, a selection operation is produced. On the other hand, the first time that a single finger is detected, the selection operation is canceled. The finger tip location relative to the pattern origin is used as the location of interest for the operation. This translates to a "mouse down" event, shown on the right in figure 7, during selection and a "mouse up" event, shown on the left in figure 7, during cancellation.

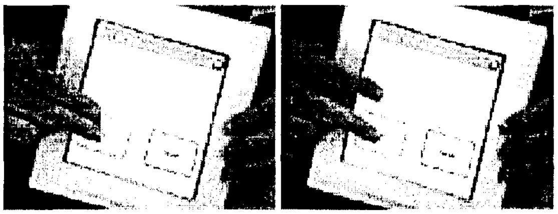

Figure 7. Control panel selection event

These events, along with the location of interest, can be sent directly to the control panel dialog box as though they were sent from the actual mouse. The purpose of using an actual dialog box in the system is to exploit the simple and powerful tools provided by the pre-existing windows programming libraries. These tools include the simplicity of adding system behaviour to this interface as well as the pre-programmed complexity of the visual interface components.

6. Importance of an augmented interface

Interaction in Virtual Reality can take on many forms. One such form is the direct manipulation of the virtual objects in the augmented environment. Another useful form is the manipulation of the system properties that govern the appearance and behaviour of the scene augmentation.

The system described in this paper is one way of providing the immersed user with the ability to control the properties of the AR system. The fact that the interface itself is a virtual object in the augmented environment allows it to be used in ways that differ from those of physical interfaces while at the same time providing complex functionality. For example, the augmented interface can be altered or positioned arbitrarily by the user or by the system. Also, the interface can be removed or replaced by other virtual objects when the system does not detect an occluding hand. In general, the virtual nature of the interface provides the user with a flexible yet powerful method of interaction.

7. Results

The system detects hand-based selection with location of interest accurate enough to manipulate common window controls. This provides the user with an effective tool for system property manipulation in an Augmented Reality environment.

The system performed proper hand extraction under reasonable lighting conditions. Drastic variations in lighting require thresholding techniques beyond those implemented in this system.

The augmentation improvements provided by the hand detection process increased immersion significantly. Inaccuracies in the visual occlusion of objects drastically hinder the user's presence with the virtual objects. Also, the instability of the virtual objects immediately separates them from the real environment. Increasing the robustness of the corner tracking technology reduced this positional variance significantly.

7.1. Performance

A major design goal of this augmented reality system is real-time frame capture and augmentation on standard PCs using off-the-shelf USB camera hardware. The current implementation of this system uses OpenGL to augment simple 2D textures onto the planar pattern at 15Hz on an Intel Pentium 3 800MHz PC equipped with an ATI Rage 128 Video card and an Intel CS110 USB camera capturing 320x240 images. A demonstration of this software is available online at www.cv.iit.nrc.ca/research/ar.

8. Conclusion

In this paper, we have demonstrated the ability to detect and respond in real-time to user interaction in an Augmented Reality environment. The importance of occlusion detection over the pattern used by the tracking system was also discussed. Adopting the commonly used windows-based interaction to the AR environment creates a familiar and comfortable interactive experience for the user.

9. References

[1] K. Dorfmuller-Ulhaas, D. Schmalstieg. "Finger Tracking for Interaction in Augmented Environments". Proceedings of IEEE and ACM Symposium on Augmented Reality, 2001. pp. 55-64.

[2] Z. Zhang, Y. Wu, Y. Shan, and S. Shafer. "Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper". Proceedings of ACM Workshop on Perceptive User Interfaces (PUI) 2001.

[3] A. Zisserman, R. Hartley. *Multiple View Geometry*. Cambridge University Press, 2000.

[4] E. Trucco, A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice-Hall, 1998.

[5] I. Shimizu, Z. Zhang, S. Akamatsu, K. Deguchi. "Head Pose Determination from One Image Using a Generic Model". Proceedings IEEE Third International Conference on Automatic Face and Gesture Recognition, April 1998. pp. 100-105.

[6] Z. Zhang. "A Flexible New Technique for Camera Calibration". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 22, No. 11, November 2000. pp. 1330-1334.

[7] P. Sturm. "Algorithms for Plane-based Pose Estimation". Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2000. pp. 706-711.

[8] J. Crowley, F. Berard, J. Coutaz. "Finger Tracking as an Input Device for Augmented Reality". Proceedings of International Workshop on Automatic Face and Gesture Recognition, Zurich, 1995. pp. 195-200.

[9] J. Rekimoto. "Matrix: A Realtime Object Identification and Registration Method for Augmented Reality". Proceedings of 3rd Asia Pacific Conference on Computer Human Interaction. pp. 63-68.

[10] H. Kato, M. Billinghurst. "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System". Proceedings of 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR) 1999. pp. 85-94.

[11] J. Molineros, R. Sharma. "Real-Time Tracking of Multiple Objects Using Fiducials for Augmented Reality". Real-Time Imaging 7, 2001. pp. 495-506.

[12] T. Auer, A. Pinz. "The Integration of Optical and Magnetic Tracking for Multi-User Augmented Reality". Computer & Graphics 23, 1999. pp. 805-808.

[13] R. Azuma, U. Neumann, S. You. "Hybrid Inertial and Vision Tracking for Augmented Reality Registration". Proceedings of IEEE Virtual Reality, 1999. pp. 260-267.

[14] A. State, G. Hirota, D. Chen, W. Garrett, M. Livingston. "Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking". Proceedings of ACM SIGGRAPH 1996. pp. 429-438.

[15] C. Harris, M. Stephens. "A Combined Corner and Edge Detector". Proceedings of 4th Alvey Vision Conference, 1988. pp. 147-151.