3D Natural Hand Interaction for AR Applications

M. Lee[1 2], R. Green[1 2], and M. Billinghurst[1]

[1] The HIT Lab NZ, University of Canterbury, Christchurch, New Zealand.
[2] Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand.

Email: {minkyung.lee, richard.green, mark.billinghurst }@hitlabnz.org

Abstract

In this paper we describe a computer vision-based 3D hand tracking system for a multimodal Augmented Reality (AR) interface. We have developed a 3D vision-based natural hand interaction method. This consists of four steps: (1) Skin colour segmentation, (2) Feature point finding, (3) Hand direction calculation, and (4) Simple collision detection based on a short finger ray for interaction between the user's hand and augmented objects. The resulting fingertip tracking accuracy varied from 3mm to 20mm depending on the distance between the user's hand and the stereo camera. We describe how this hand tracking is applied in three AR applications which merge gesture and speech input.

Keywords: Natural hand interaction, augmented reality, multimodal interface, vision-based interaction

1 Introduction

Augmented Reality (AR) technology provides a way to seamlessly overlay virtual information on a user's real environment [1]. There has been significant research in the AR field, however much of it has focused on viewpoint tracking or virtual object registration for more accurate information alignment. For example, research on markerless tracking which enables virtual information overlay on natural objects. There has been less research on interaction with the augmented virtual objects [2].

In our work we have developed a method for vision-based natural hand interaction for AR applications. When a natural hand is used as an interaction tool, there is no need to have physical markers or devices for interaction. Thus, it provides natural seamless interaction in AR environments.

In this paper we present a method for vision-based natural hand interaction with finger pointing and direct touch of augmented objects. The interface consists of four steps: (1) Skin colour segmentation, (2) Feature point finding, (3) Hand direction calculation, and (4) Simple collision detection. Three sample AR applications are described which show how our hand interaction approach can be used. In these applications, we merged gesture and speech input to provide a better user experience. There has been very little research in multimodal input for AR interfaces, and so we are keen to explore this area further.

In the rest of the paper we will first present related work (section 2), and then describe the hand tracking approach we have developed (section 3). In section 4, we show three sample applications and in Section 5 we conclude and discuss future research.

2 Related Work

We are interested in developing computer vision based multimodal interfaces for AR applications. Thus our research is related to earlier work in AR multimodal interaction.

For example, Irawati et al. [3][4] developed one of the first computer vision based AR systems with multimodal input. They extended the VOMAR application [5] by adding support for speech recognition. The final system allowed a user to pick and place virtual furniture in an AR scene using a combination of paddle gestures and speech commands. However, the system did not provide a natural free-hand interaction, instead requiring the use of a real paddle with tracking patterns on it.

Tiala and Adamo-Villani [6] developed a system which provides a polygonal modeling interface in an AR environment by using 3D Studio Max modeling tools. A sample polygonal model was overlaid on top of an ARTag [7] marker, and another ARTag marker attached to a mouse was used as a hand-held pointer device to point and to click a polygonal model. Thus, it was possible to track the mouse curser in 3D through calculating the 3D position of the ARTag marker. A user could use polygonal modeling tools in 3D Studio Max by clicking overlaid 2D icons. However, this system also did not provide natural hand interaction.

978-1-4244-2582-2/08/$25.00 ©2008 IEEE

In our research we use computer vision techniques for natural hand tracking. Piekarski and Thomas described a hand tracker based on the ARToolKit [8] for mobile outdoor environments [9]. Each ARToolKit fiducial marker was attached to the thumb and fingers of a glove, which also had metallic sensors on the fingertips, thumb, and palm to detect finger presses. The wearable system captured video by using a camera mounted on the user's head. The video was passed to the ARToolKit to track user's thumbs in world coordinates. Their work provided a way to interact with augmented object using hand gestures; however it required custom glove hardware.

HandVu is a computer vision module for gesture recognition in an AR interface [10]. It detects a user's hand in a standard pose based on texture and colour. Their "flock-of-features" tracking tracks the hand despite appearance changes. However, the output is the user's hand location in 2D image coordinates which cannot be easily used to manipulate virtual objects in 3D space.

Handy AR used a human hand as a tracking pattern to display augmented virtual objects on the hand [11]. In an offline calibration phase, a hand pose model is created by measuring each fingertip position in the presence of ground-truth scale information. A camera pose relative to the hand is reconstructed in real time. Based on the camera pose, a virtual object is augmented on top of the user's hand in 3D. However, Handy AR used the user's hand for marker-free augmentation, and not for interaction with virtual objects.

Heidemann, et al. [12] developed an intelligent system with multimodal interfaces for information retrieval in AR. It supports real-time acquiring of visual knowledge and retrieving memorized objects. An inertial sensor on the top of the user's head measures head movements. Hand gestures and speech are used to select interface menu options. Two cameras on the users head and computer vision software recognizes when they move their hand underneath the target menu and issue a voice command. Although the system has two cameras it only supports 2D menu navigation, and the speech input is used to select the menu item that the user wants, in the same way a mouse does.

As can be seen, there has been a number of AR interfaces that use computer vision based hand tracking, however none of them support 3D natural hand interaction, or are focusing on natural hand interaction for AR multimodal interfaces. In the next section we describe our approach for capturing gesture input.

3 3D Natural Hand Interaction

We have implemented a 3D computer vision-based hand tracking system, shown Figure 1. Our hand tracking is based on four steps: (1) Segmenting skin colour, (2) Finding feature points for palm centre and fingertips, (3) Finding hand direction, and (4) Simple collision detection.

We use a BumbleBee2 stereo camera from Point Grey Research Inc.[13]. It has two lenses on one camera body and provides a 640x480 pixel resolution stereo colour image with depth map at 30 frames per second. Using the camera, we can easily get the 3D information of the user's fingertip and hand centre. However, the result of disparity estimation depends on the environmental setup, such as the lighting conditions or texture of the object to calculate disparity. For easy implementation and efficient image processing, we use the OpenCV library [14]. The speech recognition module was implemented based on the Microsoft Speech API (SAPI) v5.1 [15].

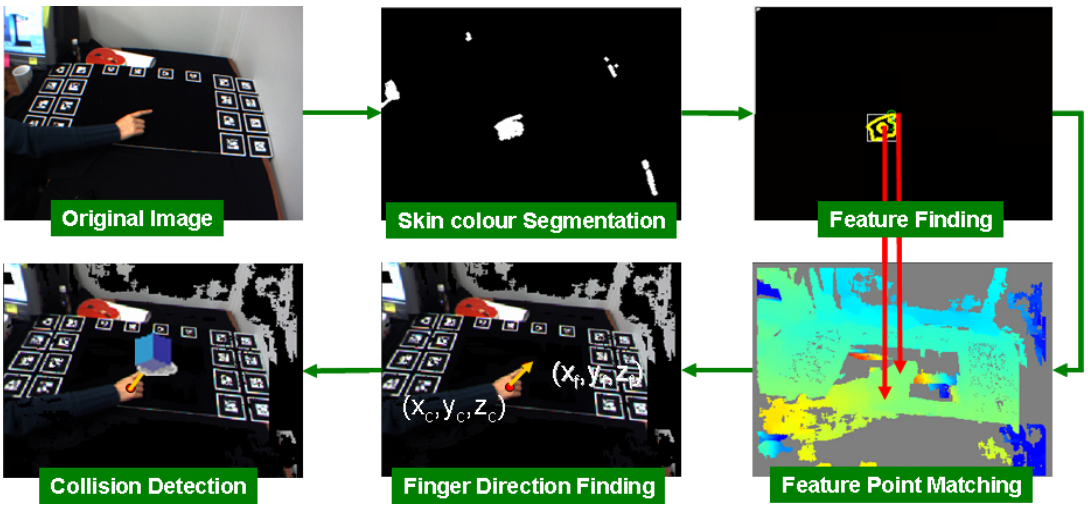


Figure 1: 3D Natural Hand Interface

3.1 Skin Colour Segmentation

To segment skin colour, we convert the camera image from RGB values to the HSV colour space. The HSV space is the most common colour space used for skin colour segmentation [16]. To find out the proper threshold value to extract only the user's hand, we use a sample skin image and its histogram of hue plane. We set the threshold boundary minimum value and the maximum value from -5 to +5 of the value which has maximum intensity. After thresholding, we perform a logical AND operation with the segmented image to get the final hand segmented image.

3.2 Feature Points Matching

Following the skin colour segmentation, we find the contour of the segmented area to identify the hand. Segmentation errors may mean that a piece of clothing or the user's face is included in the skin colour segmentation result, or the background may include objects whose hue value is included in the threshold boundary. To extract the user's hand and fingertip more accurately, we find the largest contour [17] in the thresholded output image. Afterwards, a distance transformation [18] is performed to find the centre of the palm. The farthest point inside of the contour is the centre of the palm. Next we check the convexity of the hand contour to find the candidate fingertips of user's hand. The fingertip which is the farthest from the palm is used to calculate the direction of the hand. We also find out the centre of the contour and test the convexity of the contour. As a result, we can get candidates for the fingertips of the hand. The positions of each feature point, the palm of user's hand and the fingertip, are mapped to a disparity map to estimate the 3D information of each point for the interaction with the augmented 3D object.

3.3 Hand Direction Calculation

Using these methods we can get the direction of user's hand. Calculating what user is pointing at is important for multimodal interfaces. The direction of the user's hand can be represented as a parametric equation. We have two feature points, the centre point of hand $P_c(x_c, y_c, z_c)$ and a fingertip point $P_f(x_f, y_f, z_f)$. The hand coordinate system is shown in the Figure 2.

First of all, a direction vector through P_c to P_f is calculated as:

$$(x_f - x_c, y_f - y_c, z_f - z_c) \quad (1)$$

When we pick P_c as an initial point, the line equation L from P_c to P_f can be written in a parametric form as equations (2 – 4):

$$x = x_c + (x_f - x_c)t \quad (2)$$

$$y = y_c + (y_f - y_c)t \quad (3)$$

$$z = z_c + (z_f - z_c)t \quad (4)$$

The line segment L is used to calculate where users are pointing. By specifying t, we can adjust the length of the line segment. For example, $0 \le t \le \infty$ defines the line segment whose direction is from P_c to positive infinity.

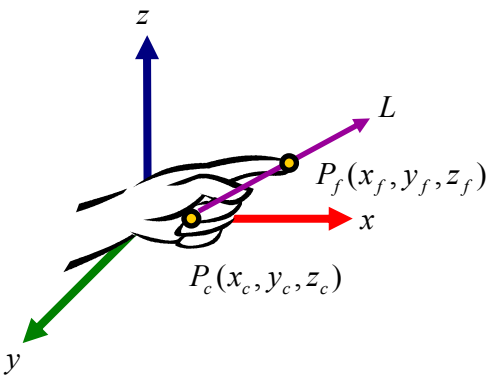


Figure 2: The hand coordinate systems P_c, P_f and pointing direction, L

3.4 Simple Collision Detection

According to Vogel's research [19], finger ray casting is fast for pointing at and selecting distant objects. If we have an infinite length ray, object selection with finger ray casting in a small working space is prone to errors by selecting all of the objects through which the ray has passed. Thus, we set the t value to $0 \le t \le 2$ which gives us a shorter ray twice the distance from P_c to P_f.

For the collision detection, we use finger ray casting, as shown in Figure 3.

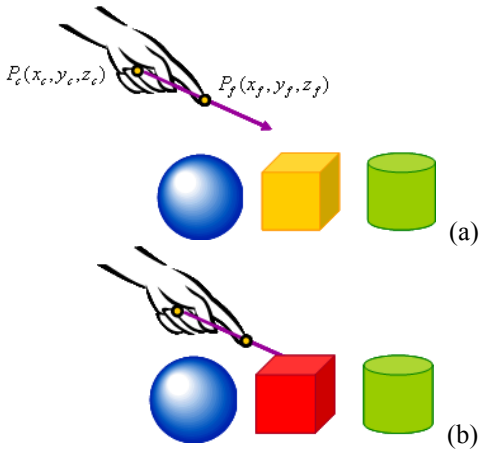


Figure 3: Collision Detection: (a) Before collision and (b) visual feedback after the collision.

In indoor AR environments, mostly augmented virtual objects are placed close to each other. Thus, collision between a finger ray of limited length and

a virtual object has to be done when the ray touches the object. We also need to provide visual or audio feedback to users to give a cue that a collision has happened.

4 Results

To show how our vision-based hand tracking system works in AR environments, we developed three different applications which used pointing or touching of augmented objects with the users' hand.

For each of these applications the user sat in front of a large 21 inch LCD monitor (Figure 4(a)) or used a Hand Held Display (HHD) for natural AR view control (Figure 4(b)).

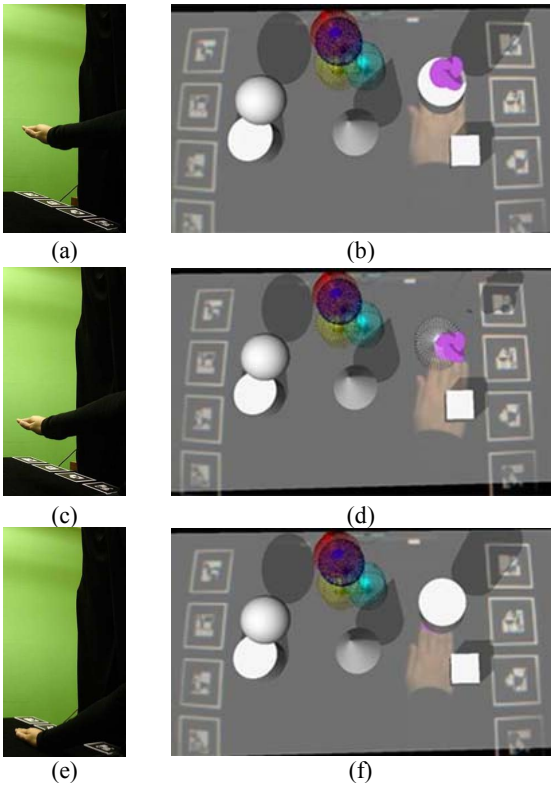
(a)

(b)

Figure 4: Tabletop AR Environment Setup (a) monitor-based and (b) HHD -based

Square tracking markers were attached to the table in front of the monitor and the table surface was used as an interaction space. The BumbleBee2 camera was mounted to the side of the table for the monitor-based AR setup and attached to the front of modified Head Mounted Display (HMD) for HHD-based AR setup. A live video view of the tabletop interaction space was shown enhanced by a virtual object overlay. We were able to track the user's fingertip with accuracy ranging from 3mm to 20mm depending on the distance between the user's hand and the stereo camera. The frame rate was 11 frames per second on a 2.4 GHz PC. The accuracy and the frame rate were enough to support real time interaction with the AR scene.

In the first application, a user could move sample objects distributed in 3D space into a final target 3D arrangement of objects (see Figure 5). The users can move their hands in all three directions to select and move objects. Figure 5 shows the system recognizing a user's hand in 3D. When the user's hand is moved within an object, then the system recognizes it as a collision (Figure 5 (c), (d)) and the object is rendered in wireframe. When the user's hand collides with an object it is selected, and then attached to the user's hand and moved along with it. Objects are deselected by moving the user's hand backward until the finger ray is far enough away not to touch the object.



(a) (b) (c) (d) (e) (f)

Figure 5: Simple application I - 3D Interaction with AR objects: different hand positions (a)(c)(e) in real environments; (b)(d)(f) in an AR environment.

In the second application, we combined gesture and speech input. The application allows the user to change the colour or shape of basic objects using speech commands. When an object is touched by the user's hand, the system provides visual feedback by rendering the object in wireframe. Speech commands were used to change the colour or shape of the objects. For example, after selecting an object with a fingertip, the user could say "colour to red" to make the object red or "change to cone" to make it a cone. The user could also change the shape and the colour of one object to the same as a second object by saying "same colour and shape as this one." after select the object to change by their fingertip. The results are shown in Figure 6.

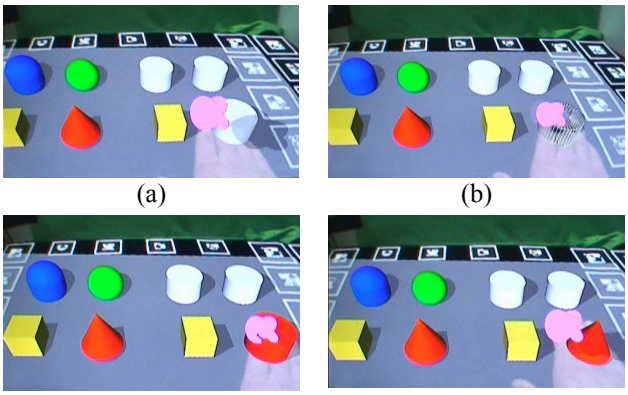


Figure 6: Sample Application II: (a) before object selection, (b) after object selection, (c) change the colour of the shape with speech input, and (d) change the shape of the object with speech input.

The third application was a scene assembly program. In this case we provided eight different virtual models that the user could use to create their own AR scene in the target area (Figure 7(a)). As shown in Figure 7(b), users could use hand gestures to stack models on top of each other. This is obviously different from 2D interfaces and provides more freedom to interact within AR environments. Users could also use speech input with their hand gestures. For example, a user could put his or her hand near the flowers and said "Daffodils". Then the yellow flowers would attach to the user's fingertip. Once attached, while the user moves his or her hand around, the flowers would follow the hand movement. Finally, when the user put his or her hand in the target area and says "drop", the flower will be dropped down on the augmented plane. Speech was also used to specify where to put the augmented virtual object, for example, by saying "Put the bird on top of the white house."

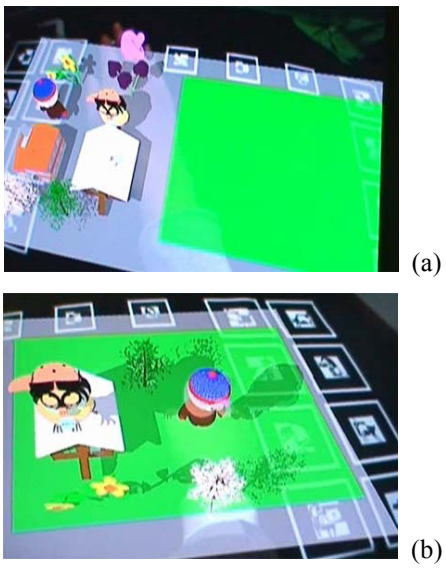


Figure 7: Sample Application III: (a) given AR scene and (b) user created AR scene.

We conducted a preliminary user study to get an idea of how users feel using natural hand gesture in an AR environment. Users reported that they felt that the natural hand-based interaction was fun to use, it supported natural movements of their hand while interacting with the virtual object, and was easy to learn how to use their hands for interaction and to select the objects. However, most of the users commented on the unstable fingertip tracking which sometimes affected their interaction.

5 Conclusion and Future Works

In this paper we have described a computer vision method for natural hand interaction with virtual objects in an AR environment. We also showed how this method was applied in three sample applications, two of which included speech input to provide a better user experience. The tracking accuracy of the hand interaction varied from 3mm to 20 mm depending on the distance between the user's hand and the stereo camera.

The proposed 3D vision-based hand interface provides users a seamless interaction in AR environments. However, using a single finger for interaction was often prone to errors because the finger was too thin to calculate the stereo disparity. Whole hand-based interaction or a higher resolution camera will be considered in the future to improve accuracy of the disparity. The system response speed is also not as fast as with data gloves or other gesture input devices. Thus, for the future work, we need to compare which interface users prefer considering naturalness, accuracy, and speed. This will be done by pursuing a user study comparing vision-based natural hand interface, to a marker-attached hand interface, and a DataGlove-worn interface.

References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. Macintyre, "Recent Advances in Augmented Reality," Computer Graphics and Applications, IEEE, Vol. 21, No. 6. pp. 34-47. 2001.

[2] J.E. Swan II and J. L. Gabbard, "Survey of User-Based Experimentation in Augmented Reality," In Proceedings of 1st international Conference on Virtual Reality. HCI International 2005, 2005.

[3] S. Irawati, S. Green, M. Billinghurst, A. Duenser, and H. Ko, ""Move the Couch Where?": Developing an Augmented Reality Multimodal Interface," In Proceedings of International Symposium on Mixed and Augmented Reality(ISMAR'06), 2006.

[4] S. Irawati, S. Green, M. Billinghurst, A. Duenser, and H. Ko, "An Evaluation of an Augmented Reality Multimodal Interface Using Speech and Paddle Gestures," In Proceedings of International Conference on Artificial Reality and Telexistence. 2006.

[5] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual Object Manipulation on a Table-Top AR Environment," In Proceedings of International Symposium on Augmented Reality (ISAR2000). pp. 111-119. 2002.

[6] P. Fiala and N. AdamoVillani, "ARpm: an Augmented Reality Interface for Polygonal Modeling," In Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR'05). 2005.

[7] M. Fiala, "ARTag, a fiducial marker system using digital techniques," In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2005), vol. 2, pp. 590- 596. 2005.

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

[9] W. Piekarski and B. H. Thomas, "Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments," In Proceedings of 1st International Augmented Reality Toolkit Workshop. 2002.

[10] M. Kölsch, M. Turk, and T. Höllerer, "Vision-Based Interfaces for Mobility," In Proceedings of International Conference on Mobile and Ubiquitous Systems (MobiQuitous). 2004.

[11] T. Lee and T. Höllerer, "Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking," In Proceedings of 11th IEEE International Symposium on Wearable Computers (ISWC2007). pp. 83-90. 2007.

[12] G. Heidemann, I. Bax, and H. Bekel, "Multimodal Interaction in an Augmented Reality Scenario," In Proceedings of International Conference on Multimodal Interfaces (ICMI'04), pp. 53-60. 2004.

[13] Point Grey Research Inc. 2008. http://www.ptgrey.com/.

[14] OpenCV Library. 2008. http://sourceforge.net/projects/opencvlibrary/.

[15] Microsoft Speech: Home Page. 2008. http://www.microsoft.com/speech/speech2007/default.mspx.

[16] X. Zhu, J. Yang, and A. Waibel, "Segmenting hands of Arbitrary Color," In Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 446-453. 2000.

[17] H. Freeman, Computer processing of line-drawing images. Computing Surveys, 6, pp. 157-97. 1974.

[18] G. Borgefors. Distance Transformations in Digital Images. *Computer Vision, Graphics and Image Processing*, 34, 344-371. 1986.

[19] D. Vogel and R. Balakrishnan, "Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays," In Proceedings of the 18th Symposium on User Interface Software and Technology. pp. 33-42. 2005.