Vision-Based Hand Detection for Registration of Virtual Objects in Augmented Reality

Kah Pin Ng, Guat Yew Tan, and Ya Ping Wong

***Abstract*—This paper presents a markerless Augmented Reality (AR) framework which utilizes outstretched hand for registration of virtual objects in the real environment. Hand detection methods used in this work are purely based on computer vision algorithms to detect bare hand without assistance from markers or any other devices such as mechanical devices and magnetic devices. We use a stereo camera to capture video images, so that the depth information of the hand can be constructed. Skin color segmentation is employed to segment the hand region from the images. Instead of fiducial markers, the center of the palm and fingertips are tracked in real time. Incorporating the depth information computed, the 3D positions of these hand features are then used to estimate the 6DOF camera pose with respect to the hand, which in turn allows the virtual objects to be augmented onto the palm accurately. This method increases the ease of manipulation of AR objects. Users can inspect and manipulate the AR objects in an intuitive way by using their bare hands.**

***Index Terms*—Augmented reality, hand detection and tracking, human-computer interaction.**

I. Introduction

Augmented Reality (AR) is an emerging technology which integrates computer generated virtual objects into the real environment. Azuma [1] introduced a widely accepted definition for an AR system as a system that combines real and virtual objects, is interactive in real time and is registered in 3D.

To register virtual objects accurately in AR system, the relative position of the camera with the scene is required. Six degrees of freedom (6DOF) that defines the position and orientation of the camera relative to the scene have to be tracked and virtual objects are then augmented in the specific position in AR environment. Some of the ways to track this information are sensor-based, vision-based and hybrid tracking techniques. Sensor-based tracking may involve sensors such as mechanical, magnetic, ultrasonic and inertial sensors. The review of the sensor-based tracking can be found in [2]. Vision-based tracking depends on the image processing techniques to compute the camera pose. Fiducial markers or light-emitting diodes (LEDs) may be added to the scene to ease the pose estimation and registration task. For example, ARToolkit library [3] uses a square with black border as marker. The marker is tracked in real time and virtual object can be augmented on the marker accurately. Hybrid tracking combines sensor-based and vision-based techniques or combines a few sensors.

Hand detection and tracking has been in growing interest as a promising way for human-computer interaction. By using hand as an interactive tool, users are able to interact with the virtual objects just like the way they interact with physical objects. The use of mechanical or magnetic sensor devices such as data gloves in the detection of hand motion is found to be more effective and less computationally expensive. However, several drawbacks arise as the use of data gloves restricts the movement of hand, requires complex calibration and setup procedures. Moreover, mechanical and magnetic devices are mostly expensive. Thus, computer vision approach which is more natural and unencumbered provides a more promising alternative to data gloves. An even more particularly desirable interaction approach is by using bare hand without the help of any markers or colored gloves. Natural features of the hand such as color, shape or other local invariant features are often used in hand detection and tracking.

In this paper, we present an AR system that uses hand for registration of the virtual objects in real 3D scene. We use computer vision techniques to detect and track the outstretched bare hand in real time. Stereo camera is used in this work, so that the 3D coordinates of the hand can be constructed for camera pose estimation.

The rest of this paper is organized as follows: Section II reviews related work. Section III describes the methodology used in this system and Section IV discusses the implementation results. Future work is included in Section V.

II. Related Work

A few researches have been carried out to utilize the hand as the tracking pattern and augmented virtual objects on top of the user's hand [4]-[8]. Users are able to inspect the virtual objects conveniently from different viewing angles without any additional markers or devices.

The systems in [4]-[6] require offline calibration for the construction of the hand coordinate system. Each user of such system must first conduct a calibration process once in order to obtain the necessary hand information. The 3D positions of hand feature points in such system are obtained. These points are then compared with the corresponding points in the captured images to obtain the 6DOF camera pose relative to the user's hand. The hand feature points used in HandyAR [4] are fingertips, detected by using

Manuscript received December 20, 2012; revised February 21, 2013. This work was supported by the Malaysian Ministry of Higher Education under Exploratory Research Grant Scheme (ERGS) with reference number 203/PMATHS/6730018.

Kah Pin Ng and Guat Yew Tan are with School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia (e-mail: nkp10_mah017p@student.usm.my, gytan@cs.usm.my).

Ya Ping Wong is with Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia (e-mail: ypwong@mmu.edu.my).

DOI: 10.7763/IJFCC.2013.V2.198

curvature-based and ellipse-fitting method. The results show that fingertip locations can be used effectively in camera pose estimation without marker-based tracking. Curvature-based fingertips detection is employed in [5] as well. In [6], the four convexity defect points between the fingers are detected and tracked instead of fingertips due to the idea that these defect points are relatively more stable during hand movement.

The presented system in [7] is an AR system on mobile devices to augment virtual objects onto the palm. The virtual objects react according to the opening and closing movements of the hand, relying on the fingertips tracking. Hand-arm segmentation is employed to find the starting points of the forearm. The four points used for camera pose estimation are constructed using the palm direction, starting point of forearm, and the convexity defect point between the thumb and index finger.

In [8], the hand-based markerless AR system aimed to be used on mobile phones. Hence, the system adopted a fast fingertip detection algorithm to enable a reasonable real time experience to the users. The performance of this method had been compared with the performance of HandyAR and results in that research show that the method used is much faster and accurate than HandyAR.

III. Methodology

The overall flow of the AR system presented in this paper is as illustrated in Fig. 1. The detailed descriptions of the methods used are included in the subsections that follow.

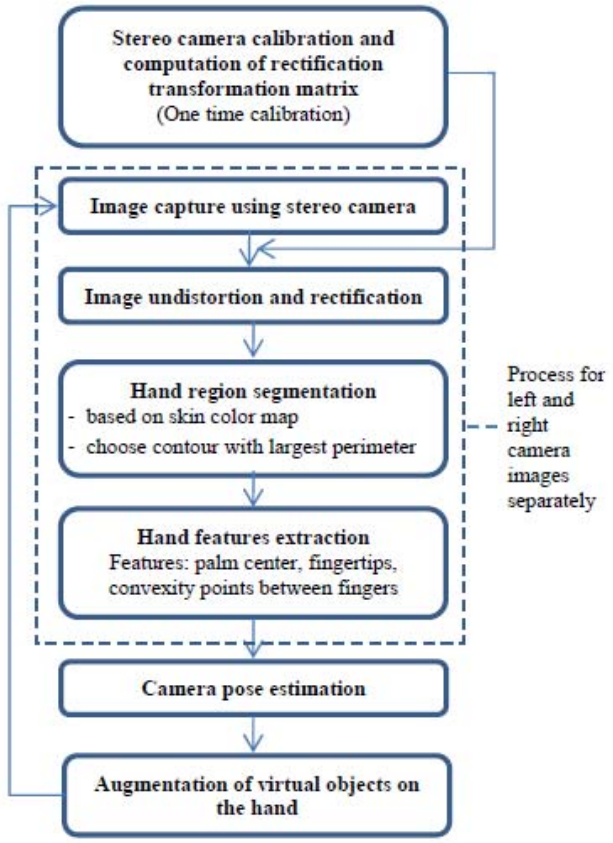


Fig. 1. Flowchart of the presented system.

A. Stereo Camera Calibration and Rectification

Offline stereo camera calibration is performed. The calibration aims to compute the camera matrices and lens distortion coefficients of both cameras. Besides, the geometrical relationship, i.e. the rotation matrix and translation vector between the two cameras in the space is obtained. Camera matrices are required for the computation of the 3D information of the captured scene and the distortion coefficients are used to remove the distortion occurred in the images.

The rectification transformation matrix is then computed. This matrix will be used for the rectification of the stereo images, so that the images will be transformed to be in row-aligned image planes. The optical axes of both cameras become parallel and so are the epipolar lines in the images. This facilitates the searching of correspondence points in both the images in the later step.

The calibration and computation of the rectification transformation matrix have to be conducted only once for each stereo camera used. The information obtained from this process is always constant for the same stereo camera. In this system, the calibration is implemented by using OpenCV library. A set of 15 paired images of a planar chessboard pattern of known size taken in different orientations and depths is being used for the calibration. After this, rectification transformation matrix is obtained. When this AR system is executed, each captured stereo images are first undistorted and rectified. The sample result of undistortion and rectification for the camera used is shown in Fig. 2.

B. Hand Region Segmentation

We adopt skin color segmentation method to first localize the skin color region in the images. This method is computational simple which allows fast implementation and thus suitable for real time applications such as augmented reality applications. In addition, this method still works despite the fact that hand, being an articulated object with high DOF, may produce a large number of possible poses and self-occlusions.

To increase invariance against illumination changes, only chrominance components of color are employed in the

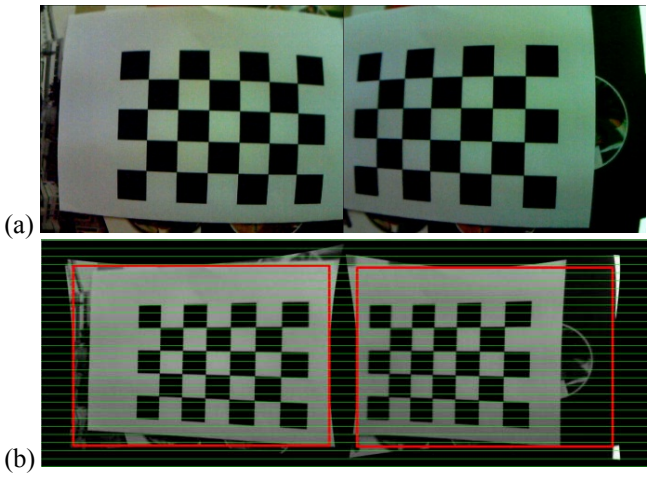
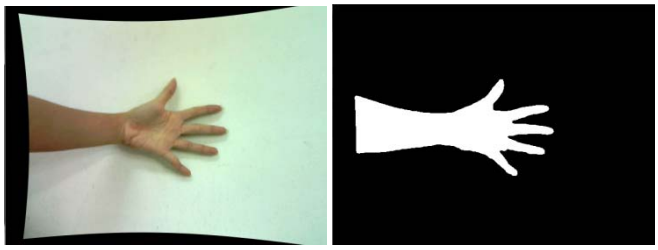


Fig. 2. Sample stereo images. (a) Before undistortion and rectification, (b) after undistortion and rectification.



(a) (b)

Fig. 3. Hand region segmentation. (a) rectified captured image, (b) possible candidate for hand region.

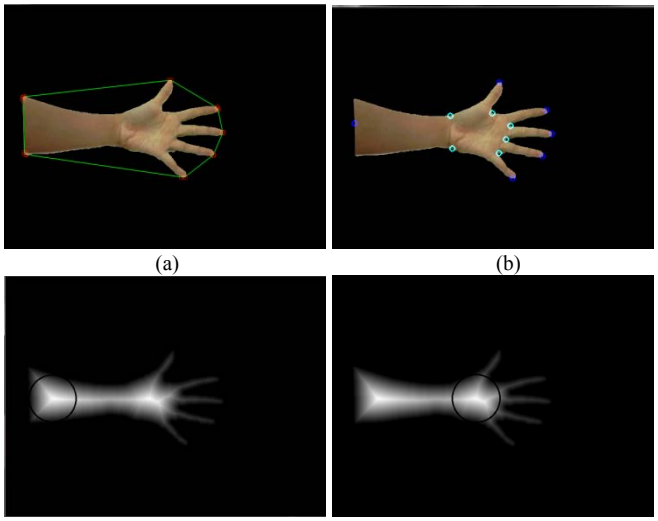
segmentation process. Thus, the color spaces which can effectively separate the chrominance and luminance components are preferable. YCbCr color space is chosen for this system. For this color space, the luminance information is contained in Y component and the chrominance information in Cb and Cr components. Each captured images are converted from RGB to YCbCr color space. Each pixel is then classified as either skin-color pixel or non-skin-color pixel based on a fixed range of skin-color map used in [8], as in (1).

$$77 \leq \mathrm{Cb} \leq 127 \text{ and } 133 \leq \mathrm{Cr} \leq 173 \quad (1)$$

The images are then converted as binary images, where skin-color pixels are converted as white pixels and non-skin-color pixels are converted as black pixels. Then, morphological operations are applied on the segmented images to eliminate small noise regions and connect adjacent large regions. Other than the hand region, there may be other skin-color objects in the background. These skin-color objects are assumed to be small and the region with the largest contour perimeter in an image is regarded as the possible hand region to be processed further. In addition, to deal with the case of which a hand may not exist in a captured image, only contour with perimeter larger than 100 pixels and area larger than 300 pixels^2 is considered to be a possible candidate for a hand region. Fig. 3 shows the result of the segmentation process to find the possible hand region in the captured image.

C. Hand Features Extraction

Next, hand features are extracted from the segmented images obtained in the previous step to further verify if a candidate hand region is indeed a hand region and to construct the coordinate system for the registration of virtual objects in the later step. The features of interest are the palm center, fingertips and convexity points between the fingers. The convex hull of the segmented region is then constructed to find the candidates for fingertips. However, there might be more than one hull point detected on a single fingertip and some false candidates detected due to the noises in segmentation, as shown in Fig. 4(a). So, the convexity defects of the contour are obtained. Any defect that has a maximum distance from the convex hull less than a threshold will be eliminated for consideration. The threshold is set to be 20 pixels through empirical observation. From the resulting defects, the average of all the hull vertices in between of subsequent defects will be regarded as the positions of fingertip. Fig. 4(b) shows the fingertips and farthest defect points between the fingers.



(a) (b) (c) (d)

Fig. 4. Hand features extraction. (a) Convex hull of hand region, (b) fingertips and farthest convexity defect points between the fingers extracted, (c) false detection of palm center if arm region is not eliminated for consideration, (d) correct detection of palm center after arm region eliminated.

Distance transform is applied on the segmented images to find the center of the palm. In most cases, the palm center is the point which is furthest away from the contour points of the hand. The exceptional case may happen when the hand turns sideway or when the upper part of arm appears in the images. So, to eliminate the arm part from consideration as palm center, the minimum and maximum x- and y-values for the convexity points are computed. The palm is assumed to be within the region specified by this x- and y-values. Thus, the pixel within this region which has the highest value from distance transform is the palm center. The center of circles in Fig. 4(c) and Fig. 4(d) indicate the palm center detected. Fig. 4(c) shows the false detection of palm center if the whole hand region is considered for the palm center while Fig. 4(d) shows the correct detection after the elimination of arm region for consideration.

D. Camera Pose Estimation

The hand is assumed to be in a plane, which the virtual objects will be registered onto it. Three correspondence points on the hand will be sufficient to establish a coordinate system for the hand. In this algorithm, the points of palm center, middle fingertip and thumb tip are used. The 2D positions of these points from the stereo images are then reprojected into actual 3D position in the real scene by using the equation as in (2).

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} \quad (2)$$

where (c_x, c_y) is principal point,
f is focal length,
d is disparity and
T_x is the horizontal shift between the cameras

All the coefficients used in the calculation above have been obtained in the camera calibration process. The 3D coordinates of the points are then (X/W, Y/W, Z/W).

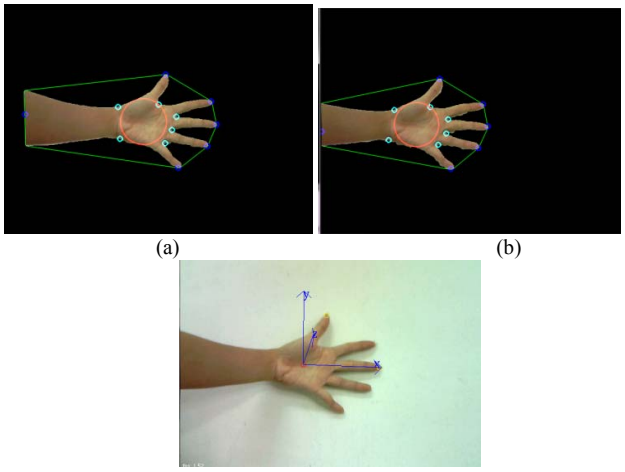


Fig. 5. Pose estimation. (a) Left image with extracted feature points, (b) right image with extracted feature points, (c) hand coordinate system established.

The palm center is defined as the origin for the hand coordinate system (HCS) while the vector from the palm center to the middle fingertip as the x-axis. The vector orthogonal to the plane with these three points, i.e. palm center (O), middle fingertip (X) and thumb tip (A) is obtained by computing the cross product of the vector $\overrightarrow{OX}$ and $\overrightarrow{OA}$. This orthogonal vector will be the Z-axis of the system. Next, the y-axis is constructed by finding the cross product of the vectors of x- and z-axis. Thus, the hand coordinate system is established.

The following step is to estimate the 6DOF camera pose with respect to the hand. A transformation matrix, which contains the translation and rotation information from the HCS to the camera coordinate system (CCS), is computed. The translation vector and rotation angles are obtained by the following steps:

1) Find translation vector, which is exactly the coordinate of the origin of HCS,
2) Translate HCS, so that the origin of HCS intersect with the origin of CCS,
3) Find rotation angle (ϕ) around x-axis such that z-axis of HCS will be on XZ-plane of CCS,
4) Rotate HCS ϕ around x-axis,
5) Find rotation angle (θ) around y-axis such that z-axis of HCS will be aligned with z-axis of CCS,
6) Rotate HCS θ around y-axis,
7) Find rotation angle (φ) around z-axis such that y-axis of HCS will be aligned with y-axis of CCS.

So, the transformation matrix from HCS to CCS can be computed as follows:

$$\left[R_z(\varphi)R_y(\theta)R_x(\phi)\,|\,T\right] \tag{3}$$

where R_z is the rotation matrix around z-axis,
R_y is the rotation matrix around y-axis,
R_x is the rotation matrix around x-axis,
T is the translation vector.

The homogenous transformation matrix is then obtained by adding a row [0 0 0 1] at the bottom of the transformation matrix. This matrix indicates the camera pose relative to the hand, which will be used to render the virtual objects in the scene. The result of camera pose estimation is as shown in Fig. 5.

IV. Implementation

The presented system has been implemented on a notebook with 2.50 GHz processor and 8GB RAM. Stereo images are captured and processed with 640x480 resolution. We use OpenCV library in the system to perform camera calibration and image processing. OpenGL is used in graphics rendering process. With the camera matrices and camera pose with respect to the hand obtained, virtual objects can be augmented on the hand. The virtual objects can be viewed in different angles by simply moving the hand into the desired pose. Fig. 6 shows a virtual cube with side 5cm, displayed on the hand in real scene. Hand is placed 30cm in depth from the camera in Fig. 6(a) and 40cm in Fig. 6(b) with almost the same pose. The virtual cubes displayed in both cases are also almost in the same pose and the sizes are relative to the hand's size. The real distance of the palm center and the middle fingertips captured in these images have been obtained as 10cm. Since the x-axes displayed in Fig. 6 are drawn with length 10cm as well, it can be observed that the system is rather accurate in reconstruction of the 3D position of the hand in real scene.

V. Future Work

For the future work, the system may include 3D interaction of the other hand of the user with the virtual objects augmented. Users are able to select and manipulate the virtual objects in an intuitive way with different hand gestures. This allows users to interact with the virtual objects just like the way they interact with physical objects and thus increase the immersive experience to the users.

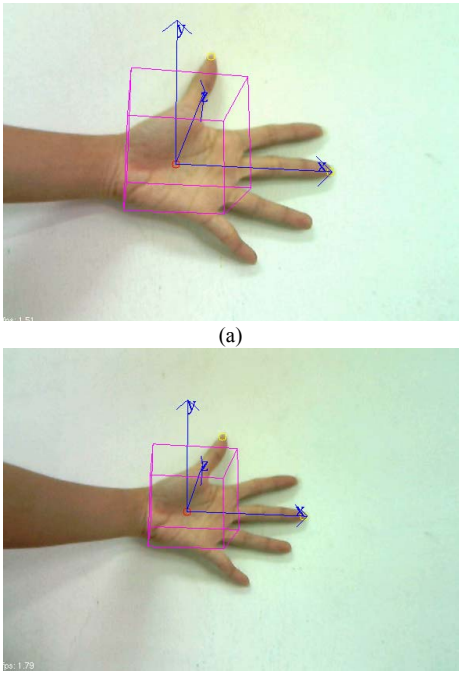


Fig. 6. A 5cm virtual cube augmented on top of the hand. (a) Hand is 30cm in depth from camera, (b) hand is 40cm in depth from camera.

REFERENCES

[1] R. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355-385, 1997.

[2] J. P. Rolland, L. D. Davis and Y. Baillot, "A survey of tracking technology for virtual environments," in *Fundamentals of Wearable Computers and Augmented Reality*, 1st ed, W. Barfield and T. Caudell, Eds. Mahwah, NJ: CRC, 2001, pp. 67-112.

[3] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. of IWAR*, 1999, pp. 85-94.

[4] T. Lee and T. Hollerer, "Handy AR: markerless inspection of augmented reality objects using fingertip tracking," in *Proc. of 11th IEEE International Symposium on Wearable Computers*, 2007, pp. 83-90.

[5] B. Lee and J. Chun, "Interactive manipulation of augmented objects in marker-less AR using vision-based hand interaction," in *Proc. of 7th International Conference on Information Technology: New Generations*, 2010, pp. 398–403.

[6] Y. Shen, S. K. Ong, and A. Y. C. Nee, "Vision-based hand interaction in augmented reality environment," *International Journal of Human-Computer Interaction*, vol. 27, no. 6, pp. 523-544, 2011.

[7] B.-K. Seo, J. Choi, J.-H. Han, H. Park, and J. Park, "One-handed interaction with augmented virtual objects on mobile devices," in *Proc. of 7th International Conference on Virtual Reality Continuum and Its Applications in Industry*, 2008.

[8] H. Kato and T. Kato, "A marker-less augmented reality based on fast fingertip detection for smart phones," in *Proc. of IEEE International Conference on Consumer Electronics*, 2011, pp.127-128.

[9] D. Chai and K. N. Ngan, "Face segmentation using skin color map in videophone applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551-564, 1999.

Kah Pin Ng graduated with Bachelor of Applied Science major in computer modeling in 2010, from Universiti Sains Malaysia (USM), Penang, Malaysia. She is a full-time research student at USM at present.

Guat Yew Tan graduated with Master of Applied Science (1996) from Nanyang Technological University, Singapore.

She is a senior lecturer at School of Mathematical Sciences, Universiti Sains Malaysia (USM), and Malaysia. Prior to her assignment in USM, she worked at the industrial sectors including consulting work in the area of Information technology.

Ms Tan is a life member in PERSAMA, Malaysia. She has written the book *C++ Programming: An Introduction*.

Ya Ping Wong is principal lecturer at Faculty of Computing and Informatics, and researcher in Vision Lab at Multimedia University, Malaysia. His research interest is computer graphics, computer vision and machine learning.