Using Watershed Transform for Vision-based Two-Hand Occlusion in an Interactive AR Environment

Peng Peng Leim, Guat Yew Tan, Kah Pin Ng, Miin Huey Ang
School of Mathematical Sciences
Universiti Sains Malaysia
Penang, Malaysia
Pengpeng_leim89@hotmail.com, gytan@cs.usm.my, nkp10_mah017p@student.usm.my, mathamh@cs.usm.my

***Abstract*—To achieve a natural interaction in augmented reality environment, we have suggested to use markerless vision-based two-handed gestures for the interaction; with an outstretched hand and a pointing hand used as virtual object registration plane and pointing device respectively. However, two-handed interaction always causes mutual occlusion which jeopardizes the hand gesture recognition. In this paper, we present a solution for two-hand occlusion by using watershed transform. The main idea is to start from a two-hand occlusion image in binary format, then form a grey-scale image based on the distance of each non-object pixel to object pixel. The watershed algorithm is applied to the negation of the grey scaled image to form watershed lines which separate the two hands. Fingertips are then identified and each hand is recognized based on the number of fingertips on each hand. The outstretched hand is assumed to contain 5 fingertips and the pointing device contains less than 5 fingertips. An example of applying our result in hand and virtual object interaction is displayed at the end of the paper.**

Keywords—watershed algorithm; two-hand occlusion; augmented reality; vision-based hand detection

I. INTRODUCTION

The key attraction in Augmented Reality (AR) is its fundamental capability to enable the virtual and real objects to co-exist in the real environment. Due to human’s adventurous nature, we do not satisfy with only the co-existence status. The urge to interact with the virtual objects in the real environment has caused further research work and resulted in interactive AR. Typically, interactive AR involves the interaction between two entities, the input device and the displayed virtual object. The discussion on displayed virtual object has been illustrated in [1]. The main focus of this paper is on the input device, particularly on the use of two-handed gestures as input and recognizing each hand by watershed transform when two-hand occlusions occur.

Hand gesture recognition is relatively new in interactive AR in which hand gestures are used to perform specific interacting tasks to control virtual objects in the real environment. Though the interactions between hands and virtual objects lack physical haptic senses, hand gesture inputs manage to provide more natural and intuitive interactions and continue to attract overwhelming attentions from researchers in many areas [2]. There are various ways to detect hand gestures. In optical-based, infrared (IR) sensors are mounted at the fingers tips and joints to detect the orientations and positions, thus the hand gestures can be determined [3]. In a contact-based approach, magnetic sensing devices for every joint of the fingers is hidden in a data glove which transmits complete signals for hand gesture recognition [4]. The accuracies in gesture detection by optical-based and contact-based inputs are high, nevertheless, expensive devices such as IR devices and data gloves are required and their attachments to the fingers limit the freedom of hand movements. In order to keep the AR set-up costs low and to overcome the hand movement limitations, marker-based input was suggested as an alternative. Marker-based input works similar to the optical-based and contact-based approach by replacing the contact and optical sensors at the fingertips and joints with fiducial markers [5]. However, the fingers look awkward with the markers attachments and they must not occlude each other in order to be detected. To achieve markerless interaction, vision-based hand gesture recognition has been explored. In vision-based interaction for AR, two hands are suggested, i.e. by using an outstretched hand for registration of 3D virtual objects, while another hand as the input device for interaction. The vision-based approach involves recording the bare hand movements with video cameras, the gestures in the images are then recognized and analysed by image processing algorithms [6]. Though vision-based approach enables bare-hand recognition, the occlusion problem remains. Two objects that are spatially separated in 3D may occlude each other in the projected 2D image plane, and causing two-handed gestures unable to be recognized directly.

A few research works have been carried out to address vision-based occlusion problems for two hands which are spatially separated [7 - 10]. In [7], the occlusion problem was avoided by fixing multiple cameras, and selecting the “best view image” by filtering out other occluded images captured. In [8], the total dissimilarity and similarity of the image was measured based on the generalized eigenvalue problem in segmenting the desired object. Reference [9] solved the crossing hands problem by tracking the arms motions with template matching, and rotating the template to find changes in the occluded and occluding hands dissimilarities, leading to the differentiation of the front and back hands. Reference [10] used EigenTracker [11] and the corresponding predictive framework [12] to solve the two-hand occlusion problem. Although the authors claimed to be able to handle all possible cases of occlusion, the result only tracked the region of interest and the hand boundary was unclear.

In this paper, we describe a method to solve the two-hand occlusion problem by using the watershed transform algorithm.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

II. Project Overview

This project is an extension to [6] which explained the steps to recognize an outstretched hand and used it for the registration of a virtual object in an AR environment. This paper adds one more hand into the scene as an interacting device to the virtual object. However, a two-handed interaction can cause mutual-occlusion problems. The main focus of this paper is to illustrate the occlusion problem we faced and how we solved it by using the watershed transform algorithm in the case of two-handed interaction.

III. Methodology

There are 6 steps to be carried out (see Fig. 1),

A. Stereo camera calibration and rectification
B. Hand region segmentation
C. Distance transform
D. Watershed transform
E. Hand feature detection for 2 hands
F. Post estimation for outstretched hand and pointing device

A. Stereo Camera Calibration and Rectification

This step removes the distortion occurred in the images and transforms the images into row aligned image planes to make the optical axes and epipolar lines of both cameras parallel. This step is carried out only once for each stereo camera used [6].

B. Hand Region Segmentation

Skin color segmentation method is used to first localize the skin region in the images. Each captured images are converted from RGB to YCbCr color space and every pixel is classified as either skin- or non-skin-colored pixel based on a fixed range of skin-color map used in [15], as in Eqn. (1)

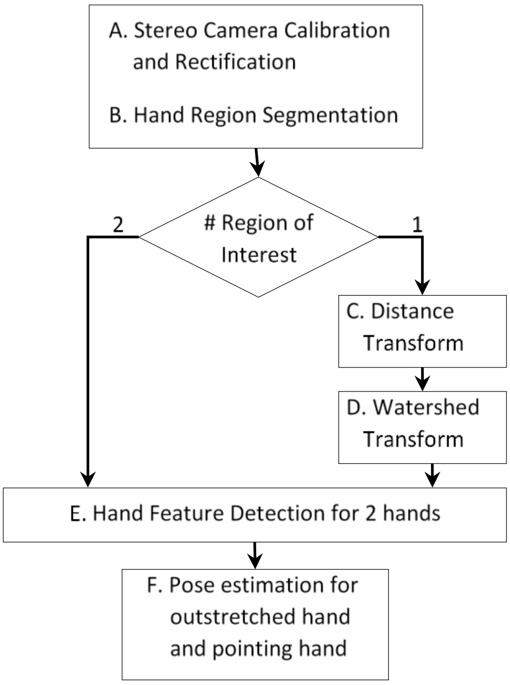

A. Stereo Camera Calibration and Rectification
B. Hand Region Segmentation
# Region of Interest
2
1
C. Distance Transform
D. Watershed Transform
E. Hand Feature Detection for 2 hands
F. Pose estimation for outstretched hand and pointing hand

Fig. 1. The algorithm framework.

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

Eventually, skin-colored and non-skin-colored pixels are converted to white and black pixels respectively to form a binary image. Next, morphological operations are applied to the segmented image to eliminate the small noise regions and connect the adjacent large regions. Other than the hand region, there might be other skin-colored objects in the background. However, these skin-colored objects are assumed to be insignificant and will not be considered. Only the region with the largest contour parameter is defined as a possible hand region.

It is also possible that the hand does not appear in the image but instead noise is captured. To avoid applying recognition processes to non-existent objects, contours with perimeter larger than 200 pixels and areas larger than 500 pixels will be considered as possible hand regions. When two largest contours are found, we assume that the two hands are not mutually occluded. When there is only one large contour in the image, both hands are assumed to occlude each other and will proceed for distance transformation and watershed algorithm, to segregate the contour of the two hands. In both cases, hand feature extraction is performed to distinguish a hand as an interacting tool and another hand as a panel to display virtual object. Fig. 2 shows the result of the segmentation process in finding the possible hand region in the captured image. Fig. 2 (a) and (b) show the input images for the two-handed occlusion and non-occlusion cases respectively; Fig. 2 (c) and (d) show the output.

C. Distance Transformation

Distance transform has been used to locate the peaks of the objects. Basically, it counts the distance of each non-object pixel (black) to the object pixel (white), and converts a binary image to a grey scale image with the grey tones running from the boundary to the center of the object based on the distance values.

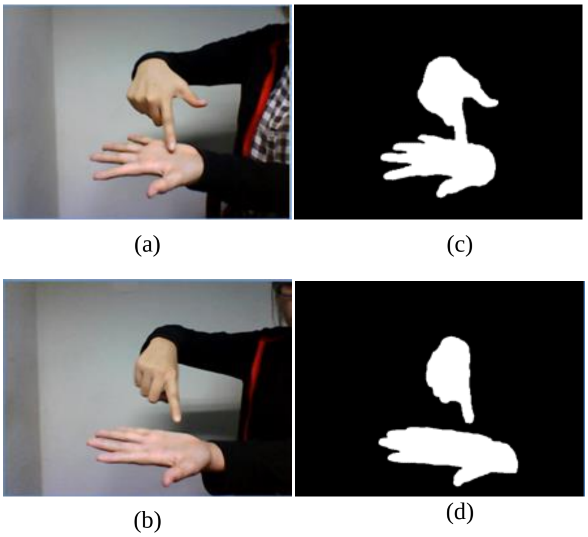


Fig. 2. Skin Color Segmentation for two cases. The input images for the 2-hand occlusion and non-occlusion cases are shown in (a) and (b), respectively; the output of the segmentation are shown in (c) and (d), respectively.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

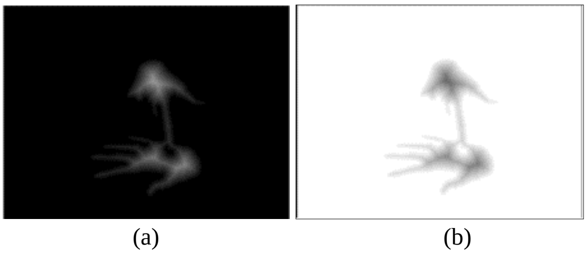


Fig. 3. Chessboard distance transform result using Fig 2(c) as input image. The distance of each pixel is shown in (a), the negation of (a) is shown in (b).

In our project, we use chessboard distance transform for its speed and to prevent over-segmentation [16]. From the previous steps in B, distance transform will apply to the image segments if there is one large contour found in the image. Fig. 3 shows the result of distance transform by using Fig. 2(c) as an input image. Fig. 3(a) shows the distance of each pixel, running from white to black, with white being the pixels with zero distance to the object and black being the farthest from the object. Fig. 3(b) shows the negation of Fig. 3(a) for the watershed algorithm to be applied in the next step.

D. *Watershed Transformation*

Watershed is an algorithm based on the concept of topographic representation of negation of distance transform map (see Fig. 3(b)). The greater the pixel values, the higher gradient of ridge area in the image; while low gradient pixels are considered as the basin. The idea of the watershed algorithm is like a water stream falling from the ridge. It flows along a path to reach the catchment basin to form a region. The watershed lines divide the catchment sink into different areas which form the object contours in the image. Fig. 4 shows two regions of interest which are segmented to different colors.

E. *Hand Feature Detection for Two Hands*

To verify both regions as hands and further to distinguish the regions as the outstretched hand and the pointing hand, hand-feature extraction is performed. The features of the outstretched hand are fingertips and palm center. To extract the fingertips, the convex hull and convexity defect algorithms are used. In OpenCV, the convexity defect function identifies all points including both convex and concave points (see Fig. 5). The angle of every three consecutive points in the convexity defect is calculated. If the angle is less than 80 degrees, the middle point is checked against the set of points obtained in convex hull, and if the middle point is found as one of the convex hull vertex, the point is detected as a fingertip. Both contour regions are examined, and the results are displayed in Fig. 6.

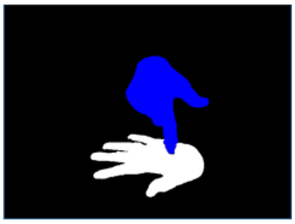


Fig. 4. Two regions of interest after watershed transformation.

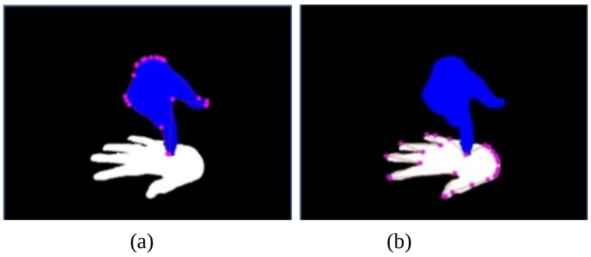
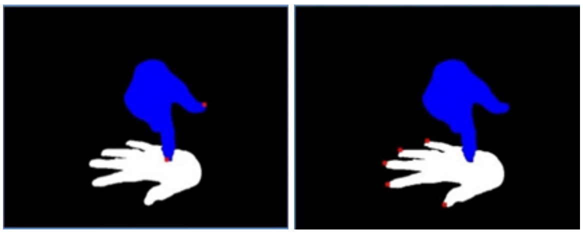


Fig. 5. Convex hulls and convexity defects of each contour region (a) and (b) are highlighted as purple dots.



(a) Result of Fig. 5 (a)

(b) Result of Fig. 5(b)

Fig. 6. Fingertips of each hand region after hand feature detections. Result of Fig. 5(a) is shown in (a). Result of Fig. 5(b) is shown in (b).

The region which consists of less than 5 points is considered as the pointing hand and the other as the outstretched hand.

F. *Pose Estimation for the Outstretched Hand and the Pointing Hand*

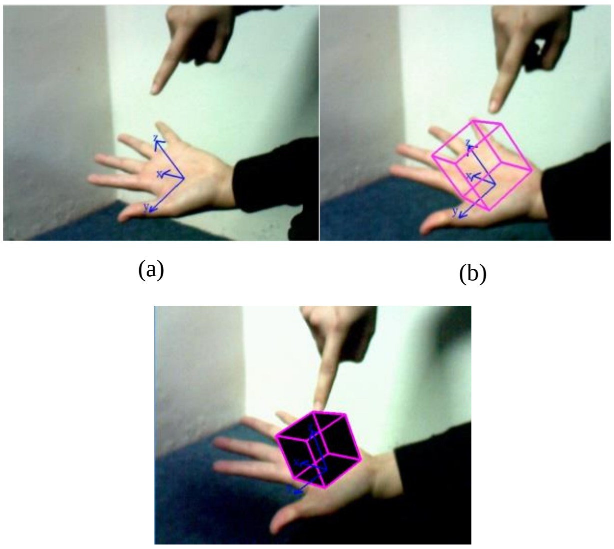
From the previous step, the region with five fingertips is considered as the outstretched hand which serves as the plane for virtual object registration. To form a 3D hand coordinate reference frame, the middle fingertip, the palm center (found by center of mass algorithm) and the thumb tip form a right angle with the palm center as the origin, O; the middle and thumb as the X- and Y- axes respectively. The Z-axis is calculated by taking the cross product of the OX and OY vectors. The camera pose with respect to the hand is then estimated by using the transformation matrix defined in [6].

For both hands, the 2D positions of the points obtained in step E are re-projected onto the actual 3D positions in the real scene by using the following equation:

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

where (cx, cy) is the principal point,
f is the focal length,
d is the disparity, and
Tx is the horizontal shift between the cameras.

All the coefficients above have been obtained in the camera calibration process. The 3D coordinates of the points are then (X/W, Y/W, Z/W) [6].



(a) (b)

(c)

Fig. 7. Separate hands showing virtual objects augmentation on the outstretched hand and pointing hand interaction with virtual cube. (a) Hand axes augmentation. (b) Hand axes and cube augmentation. (c) When the pointing finger "touches" the virtual cube, the wire cube is changed to solid cube.

IV. RESULTS

For separated hands, the hands' positions and finger-object interaction are displayed in Fig. 7, where Fig. 7(a) and (b) show the hand coordinates with the hand axes and virtual cube displayed on the outstretched hand. Fig. 7(c) shows the interaction between the pointing hand and the virtual cube. The fingertip of the pointing hand has "touched" the virtual cube to change the wire cube to a solid cube. At this stage, only one pointing gesture is recognized.

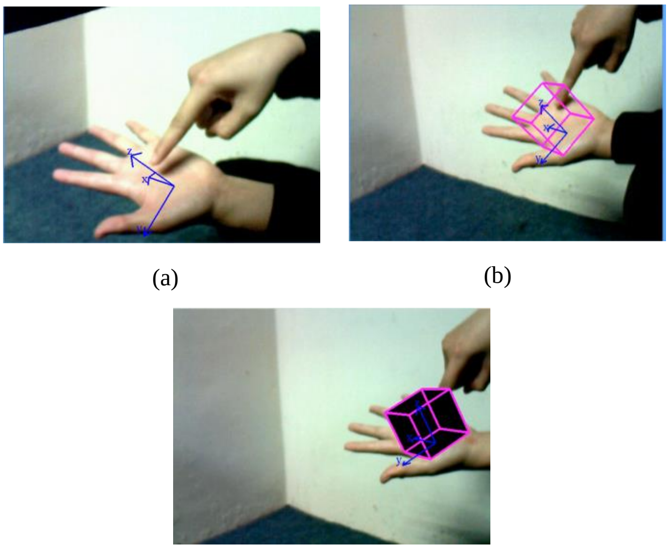
Fig. 8 shows the two-hand occlusion images, where both outstretched and pointing hands can be detected separately.

V. TECHNICAL ENVIRONMENT

The 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 the OpenCV library to perform camera calibration and image processing. OpenGL is used in the graphics rendering process. With the camera matrices and camera pose with respect to the hand obtained, virtual objects can be augmented on the hand.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we presented a vision-based methodology to extract two-handed gesture by using the watershed transform in a two-hand occlusion for interactive AR. The coordinates of both hands are calculated, for virtual objects augmentation and finger-object interaction. In our project, bare hands are used as inputs. At this stage, we are able to recognize one pointing gesture which uses one finger only and to change the wire cube to solid upon touching. In the future, we endeavor to recognize more gestures and incorporate more finger-object interactions, e.g. rotating and translating the object by the pointing finger.



(a) (b)

(c)

Fig. 8. Outstretched hand and pointing hand are identified separately in an occlusion case. Hand axes and the virtual cube are augmented on the outstretched hand, (a) and (b). Pointing finger "touches" the virtual cube, the wire cube is changed to solid cube (c).

ACKNOWLEDGMENT

This work is supported by the Short Term Grant Universiti Sains Malaysia (304/PMATHS/6312096). This paper is an extended version of our paper presented at the 9th International Conference on Information Technology and Applications (ICITA), in Sydney, Australia.

REFERENCES

[1] P. P. Leim and G. Y. Tan, "Component level interaction of a 3D model in an interactive augmented reality environment," International Journal on Future Computer and Communication, vol. 2, no. 5, pp. 539–542, October 2013.

[2] H. S. Hasan and S. A. Kareem, "Human computer interaction for vision based hand gesture recognition: a survey," Proceedings of the IEEE International Conference on Advanced Computer Science Applications and Technologies (ACSAT), 2012, pp. 55–60.

[3] S. Reifinger, F. Wallhoff, M. Ablaßmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," International Conference on Human-Computer Interaction, C. Stephanidis, Eds. Beijing: Springer, July 2007, pp. 728–737.

[4] J. D. Sturman and D. Zeltzer, "A survey of glove-based input," Computer Graphics and Applications, IEEE, vol.14, no. 1, pp. 30–39, 1994.

[5] V. Buchmann, "FingARtips: gesture based direct manipulation in Augmented Reality," in Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques, Australasia and South East Asia, ACM, 2004, pp. 212–221.

[6] K. P. Ng, G. Y. Tan and Y. P. Wong, "Vision-Based Hand Detection for Registration of Virtual Objects in Augmented Reality," International Journal of Future Computer and Communication, vol. 2, no. 5, pp. 423–427, October 2013.

[7] A. Utsumi and J. Ohya, "Multiple Hand Gesture Tracking using Multiple Cameras," IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 1999, pp. 473–478.

[8] J. Shi and J Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.

[9] T. Inaguma, H. Saji, and H Nakatani, “Two-handed gesture tracking in the case of occlusion of hands,” in IAPR Workshop on Machine Vision Applications, 2002, pp. 306–309.

[10] K. A. Barhate, K. S. Patwardhan, S. D. Roy, S. Chaudhuri, and S. Chaudhury, “Robust two hand tracker using predictive eigentracking,” in Proceedings of the National Conference on Communication (NCC), 2004, pp. 101–105.

[11] J. M. Black and A. D. Jepson, “Eigentracking: Robust matching and tracking of articulated objects using a view-based representation,” International Journal of Computer Vision, vol. 26, no.1, pp. 63–84, 1998.

[12] N. Gupta, P. Mittal, S. D. Roy, S. Chaudhury, and S. Banerjee, “A Predictive Scheme for Appearance-based Hand Tracking,” Proceedings of the National Conference on Communications (NCC), 2002, pp. 513–522.

[13] S. Beucher, “The watershed transformation applied to image segmentation,” Scanning Microscopy Supplement, pp. 1–26, 1992.

[14] L. J. Belaid and W. Mourou, “Image Segmentation: A Watershed Transformation Algorithm,” Image Analysis & Stereology, vol. 28, no. 2, 2009, pp. 93–102.

[15] H. Kato and T. Kato, “A marker-less Augmented Reality based on fast fingertip detection for smart phones,” IEEE International Conference on Consumer Electronics (ICCE), 2011, pp. 127–128.

[16] Q. Chen, X. Yang, and E. M. Petriu, “Watershed segmentation for binary images with different distance transforms,” Proceedings of the 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications, 2004, pp. 111–116.