# A Hand Gesture Control Framework on Smart Glasses

Chih-Hsiang Yu[1], Wen-Wei Peng[1], Shys-Fan Yang-Mao[1], Yuan Wang[2], Winyu Chinthammit[2], Henry Been-Lirn Duh[2]
[1] Industrial Technology Research Institute
Taiwan
{SeanYu, WaynePeng, YangMao}@itri.org.tw
[2] Human Interface Technology Laboratory Australia
University of Tasmania, Australia
{Yuan.Wang, Winyu.Chinthammit, Henry.Duh}@utas.edu.au

## Abstract

In this paper, we proposed a hand gesture control framework on smart glasses. Three different camera structures were presented to detect the hand portion, and the Moore's Neighbor tracing algorithm detects the hand contour more efficiently and automatically. We not only refined the skin-color model but also improved the Chamfer matching method for the robust and effective gesture recognition.
A demonstration has been implemented by using the hand gesture control framework. Several gestures are pre-defined for various functions, such as selecting a virtual 3D object, rotating, zooming in or zooming out, and changing display properties of the 3D object.

**CR Categories:** I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.5.4 [Pattern Recognition]: Applications—Computer Vision;

**Keywords:** hand gesture, smart glasses, gesture recognition

## 1 Introduction

Nowadays, in order to overcome limitations of WIMP interaction, many novel emerging user interfaces have been discussed, such as multi-touch user interfaces [Reisman et al. 2009], tangible user interfaces (TUIs) [Jordà et al. 2007], organic user interfaces (OUIs) [Koh et al. 2011], and mid-air gesture detection [Baudel and Beaudouin-Lafon 1993; Benko and Wilson 2010]. These technologies have the potential to significantly impact on marketing in the area of smart TVs, desktops, mobile phones, tablets and wearable devices such as smart watches and smart glasses. As we know, Google Glass, a type of wearable device, which only provides a touch pad, located on the right side of the device, which can use touch gestures by simple tapping and sliding your finger on it. Hand gesture is not only one of powerful human-to-human communication modalities [Chen et al. 2007], but also can change the way with human-computer interaction. Therefore, implementing a hand gesture control framework on the glasses could provide an easy-to-use, intuitive and flexibility of interaction approach.


*SA'15 Symposium on MGIA* , November 02-06, 2015, Kobe, Japan
ACM 978-1-4503-3928-5/15/11.
http://dx.doi.org/10.1145/2818427.2818444

In this paper, we proposed a hand gesture control framework on smart glasses that supported various fancy gesture controls. The user can load a virtual 3D object through his fingers just like the magician's trick; rotate the virtual 3D object by moving his hand; zoom the virtual 3D object by using a particular gesture sign.

## 2 Our Approach

Our framework uses a camera to capture the image stream, and by image processing it can recognize user's gestures and generate interactive events to update the user interface.

### 2.1 Framework Overview

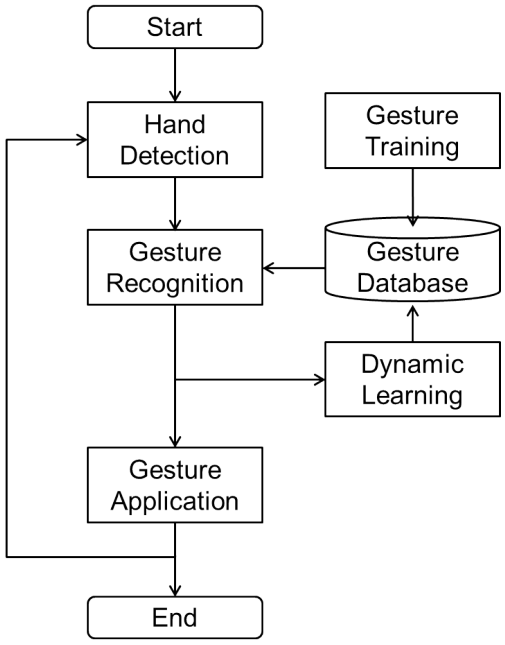


**Figure 1:** *Framework Overview.*

Figure 1 illustrates our framework overview. After camera capturing the image steam, the system detects the hand features, including the hand contour, the hand center and the hand size for the gesture recognition. According to the hand features the system transforms each matching template from the gesture database and calculates the matching cost. The template with the minimum matching cost is chosen as the recognized gesture. By learning the recognized gestures the system can dynamically adjust the upper-limit threshold of the template matching for the stable gesture recognition with different users. Furthermore, by combining the recognized gesture with the hand moving detection, the system generates different controllable events for the user interface.

## 2.2 Hand Detection

To detect the hand features, we need to identity the hand portion, in other words, to eliminate the background. There are various types of build-in cameras with smart glasses. According to different types of glasses, we present how to detect the hand portion of the user with different camera structures, including color camera, IR camera and depth camera.

### 2.2.1 Detect Hand Portion

In the case of the color camera, we at first detect the rough skin-color portion and then refine the personal skin-color model of the current user. There are several skin-color models by statistics

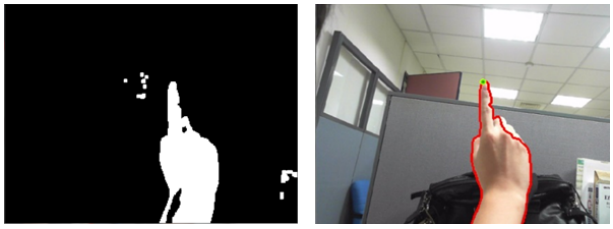

**Figure 2:** *The refined skin-color model and the hand portion by using color camera.*

[Garcia et al. 1999; Hsu et al. 2002], and in this paper we use the nonlinear transformation of Chroma and the skin model presented by Hsu et al. [Hsu et al. 2002]. Since this skin-color model has higher flexibility in contrast with the brightness change of the environment light. With the rough skin-color model, we further use the Gaussian model to refine the skin-color for the current user. The Gaussian formula can be presented as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where $x$ is the pixel's color vector inside the hand region, $\mu$ and $\sigma$ are the average and the standard deviation of these color vectors. The Gaussian $f(x)$ computes each pixel's probability of the hand portion from the camera image, and the maximum area of connected component of these pixels is determined as the hand portion of the current user. The result is shown in Figure 2.

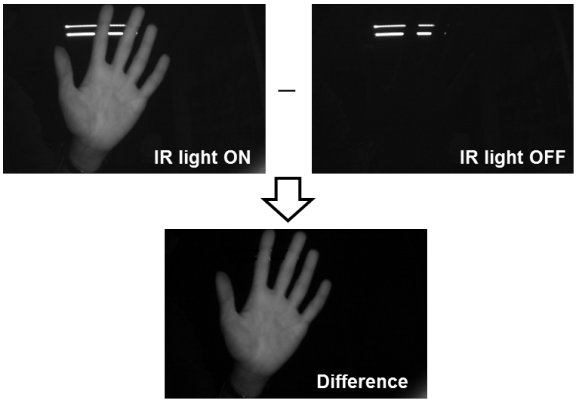


**Figure 3:** *Take the different image regions between the images with IR lighting ON and OFF as possible hand region.*

In the case of the IR camera, we take the different image regions between the images with IR lighting ON and OFF as the possible hand region (Figure 3). This method can eliminate the high brightness light at the background and extract the nearby objects in front of the IR camera. The maximum area of connected component of these regions is determined as the hand portion.

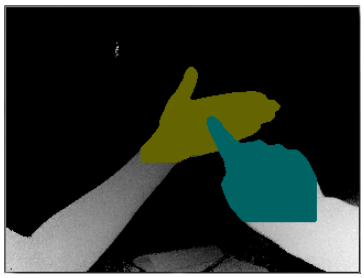

**Figure 4**: *3D connected component method for object separation.*

In the case of the depth camera, we at first separate image objects by using the 3D connected component as shown in Figure 4. We take the objects with the nearest distance from the camera and with enough region size as the hand portion.

### 2.2.2 Detect the Hand Contour

To detect the hand contour, we trace the edge of the hand portion more efficiently and automatically by using Moore's Neighbor tracing algorithm. Moore's Neighbor tracing algorithm defines M (p) to be the Moore neighborhood of current pixel p and outputs a sequence B (b1, b2, ..., bk) of boundary pixels i.e. the contour. Given a detected hand portion (i.e. Figure 2 shows the result of the hand contour detection), the system detects the hand contour by using Moore's Neighbor tracing algorithm.

### 2.2.3 Detect the Hand Center and the Hand Size

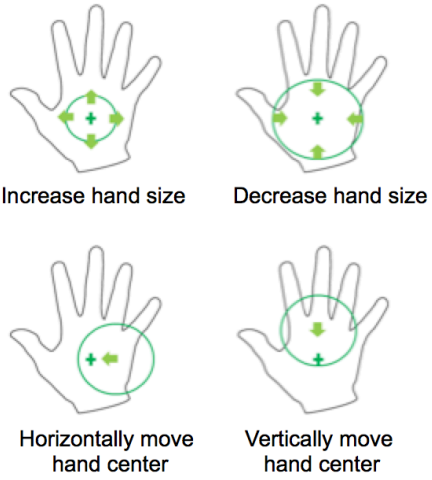


**Figure 5:** *The maximum inscribed circle of hand portion.*

We use the detected hand portion to detect the hand center and the hand size by calculating the maximum inscribed circle of the hand portion. We set the center of the hand contour as the hand center at initialization, and take the hand center of the previous frame as the start searching point at runtime. According to the position of the start searching point, we expand the circle radius to reach the contour and dynamically adjust the circle center until finding the maximum inscribed circle of the hand region as shown in Figure 5.

While the maximum inscribed circle is calculated, the center and the radius of the maximum inscribed circle are represented as the hand center and the hand size. In experience, we set the maximum iteration number as 20 to stop the calculation loop.

## 3.3 Gesture Recognition

For robust and effective gesture recognition, we take the result of the hand features detection to speed the time of template matching. Our template matching method is based on the Chamfer distance transform (Chamfer DT). Since the Chamfer DT has higher flexibility against the matching error, we demonstrate the traditional matching method by using the Chamfer DT and our improved Chamfer matching method as follows.

### *3.3.1 Chamfer DT*

Since the Chamfer DT preserves the edge features for the space correlation, we take the detected hand contour as an input to calculate the Chamfer DT image. At initialization we set the pixels of the hand contour as value 0 and other pixels as value 255 to start the two pass operations of the Chamfer DT. The two pass operations of the Chamfer DT can be represented as follows:

The result of the Chamfer DT preserves the pixels of the hand contour with value 0 and the pixels nearby to the hand contour with a low-value as shown in Figure 6. The property of the Chamfer DT gives flexibility for the template matching against some influence of the matching alignment complexity, including scale, translation and rotation.

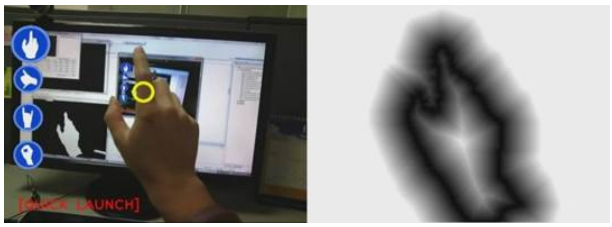

**Figure 6:** *The result of Chamfer DT for hand image.*

### *3.3.2. Chamfer DT Matching*

The matching complexity is dependent on scale, translation and rotation between the template and the detected hand portion. To reduce the matching complexity, the system calculates the matching cost based on the detected hand center and hand size. Figure 7 illustrates our improved Chamfer matching method. We first scale the template image so that the hand sizes between the detected hand and the matching template are consistent. Then by aligning the two hand centers in matching images, the system can efficiently calculate the pixel differences. Since the fingertip-up is a different gesture from fingertip-left or fingertip-right, we do not have to rotate each angle of the hand. Furthermore, the Chamfer DT has flexibility for the rotation of DOF (degree of freedom), so the matching complexity can be reduced.

Improved by the Chamfer matching, we can reduce the time complexity from original O(n*n) to O(1). In addition, it can preserve high recognition accuracy. The advantages of the method are fast and robust for even more templates.

To calculate the matching cost between the detected image and the template image, we use the peak signal-to-noise ratio (PSNR) to find the template with the minimum cost. Figure 8 shows that even with several noises in the contour, the Chamfer matching method still can recognize the correct gesture.

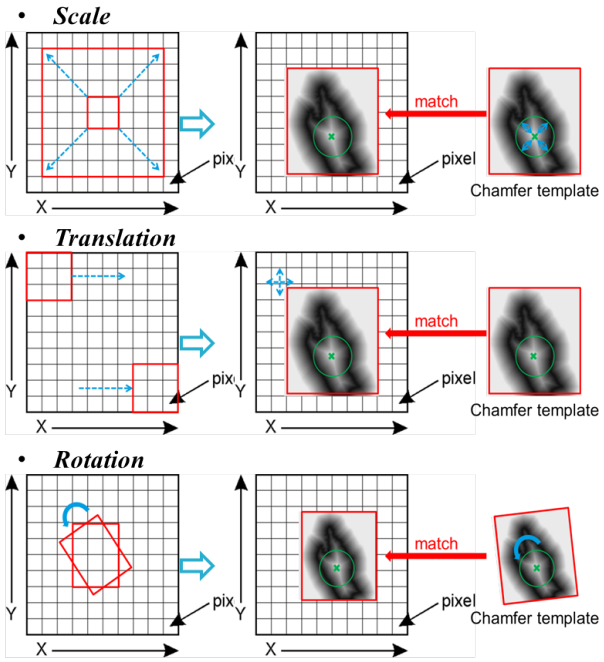


**Figure 7:** *Our improved Chamfer matching method against scale, translation and rotation complexity.*

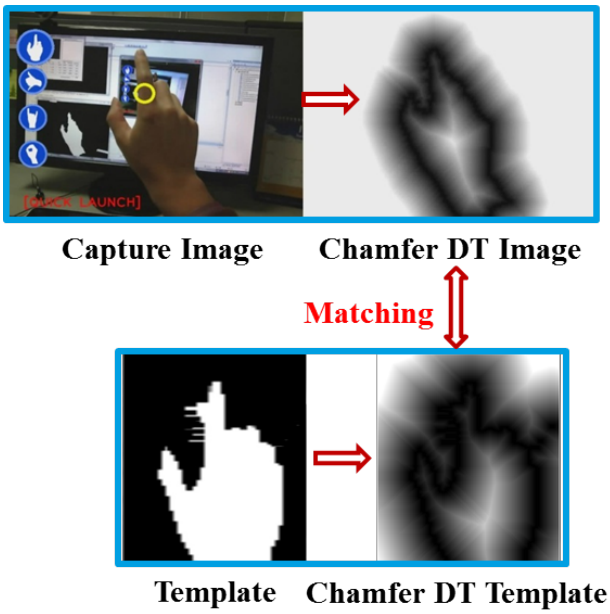


**Figure 8:** *Chamfer DT matching between detected hand and templates.*

### 3.4. Gesture Application

Figure 9 illustrates our defined gestures. To reduce the error-action of hand interaction, we define the gesture events by compositing the static gestures and hand motions. The system needs to detect the start gesture for 0.1 sec, so the system will unlock for triggering gesture events. In addition, if the unlock state is over 2 sec and there is no start gesture within this 2 sec, the system will re-lock the system.

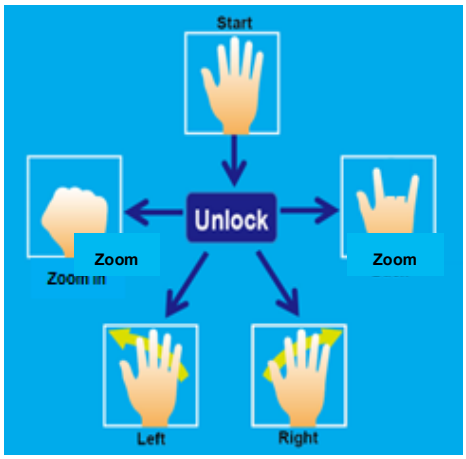


**Figure 9:** *Our defined gestures.*

The four defined gesture events are Left, Right, Zoom in and Zoom out. The Zooming event is composited by two static gestures. The Left and Right events are decided after hand moving exceeds the defined horizontal threshold with enough moving speed.

## 4 Demonstration

We implement a demonstration by using the hand gesture control framework. As shown in Figure 11 and Figure 12, a 3D object can be manipulated by using hand gesture. To rotate the 3D object, simply move the hand to the left or to the right on horizontal axis. To zoom the 3D object, just close the hand or make an American sign Language gesture for "I Love you".

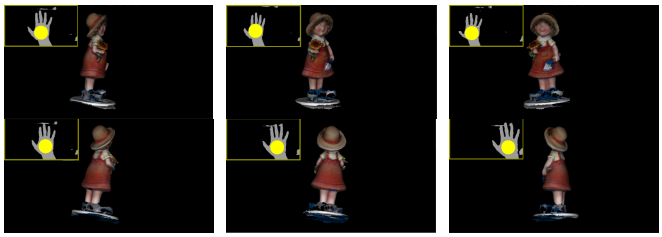

**Figure 11:** *Rotating 3D Model.*

**Figure 12:** *Zooming 3D Model.*

## 5 Conclusion

The proposed framework has been implemented and the framework provides a series of pre-defined gestures on smart glasses. It is our hope that the framework is easy to use and convenient for developers to create applications with hand gesture controls. Currently, we have used single hand gestures, and we are going to add more gestures into the database and make two hands gestures as well. We will focus on the usability test and the accuracy on this framework in the next step. The hand gestures control technologies may be carried out for the smart glasses in the future.

## References


BAUDEL, T., AND BEAUDOUIN-LAFON, M. 1993. Charade: remote control of objects using free-hand gestures. *Communications of the ACM*, *36*(7), 28-35.

BENKO, H., AND WILSON, A. D. 2010. Multi-point interactions with immersive omnidirectional visualizations in a dome. In *ACM International Conference on Interactive Tabletops and Surfaces,* ACM Press , 19-28.

CHEN, Q., PETRIU, E. M., AND GEORGANAS, N. D. 2007. 3D hand tracking and motion analysis with a combination approach of statistical and syntactic analysis. In *Haptic, Audio and Visual Environments and Games, HAVE 2007, IEEE International Workshop,* 56-61.

GARCIA, C., ZIKOS, G., AND TZIRITAS, G. 1999. Face detection in color images using wavelet packet analysis. In *Multimedia Computing and Systems, IEEE International Conference on* Vol. 1, 703-708.

HSU, R. L., ABDEL-MOTTALEB, M., AND JAIN, A. K. 2002. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(5), 696-706.

JORDÀ, S., GEIGER, G., ALONSO, M., AND KALTENBRUNNER, M. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction,* ACM, 139-146.

KOH, J. T. K. V., KARUNANAYAKA, K., SEPULVEDA, J., THARAKAN, M. J., KRISHNAN, M., AND CHEOK, A. D. 2011. Liquid interface: a malleable, transient, direct-touch interface. *Computers in Entertainment (CIE)*, *9*(2), 7.

REISMAN, J. L., DAVIDSON, P. L., AND HAN, J. Y. 2009. A screen-space formulation for 2D and 3D direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ACM, 69-78.