# A 3D Interaction Technique Based on Gesture Recognition

**Keyu Wang, Yang Li, Hao Fu, Chao Ping Chen*, Bing Yu**

*Smart Display Lab, Shanghai Jiao Tong University, Shanghai, China*

*Email: ccp@sjtu.edu.cn*

## ABSTRACT

*We propose a three-dimensional interaction based on the hand gesture recognition. Two machine learning algorithms are employed. Our results show that the support vector machine outperforms the k-nearest neighbors in terms of accuracy. The accuracies of support vector machine are 90.9% and 72.6% in the simple and complex backgrounds, respectively.*

## INTRODUCTION

With the development of 3D display technology, smart glasses are expected to become the next-generation augmented reality (AR) platform. As the traditional input devices such as mouse, keyboard and touch panel are no longer equipped on smart glasses, computer-vision-based hand gesture is considered as the proper interaction technique for smart glasses. The traditional methods of hand gesture recognition rely on the color grayscale cameras [1,2]. Suffering from varying illuminations, those methods usually have a low accuracy in hand tracking. Recently, low-cost depth camera such as Kinect is being widely adopted in the consumer electronics, which provides accuracy hand tracking for hand gesture interaction [3]. However, most of those researches [4,5] are carried on the complex instruction set computer, e.g. x86-based computers. For mobile devices such as smart glasses, a lower power consumption processor, e.g. reduced instruction set computer, is preferred. In this paper, we propose a 3D interaction technique based on reduced complexity gesture recognition algorithm which is applied to ARM-based smart glasses [6].

Our 3D interaction technique is composed of 3D display and gesture recognition. Virtual 3D objects are first rendered by Unity, and then projected by the smart glasses to be overlaid with the real world. The hand gesture image sequences including depth and color information are captured by a built-in binocular stereo camera. We combine skin-color model and depth image to segment the hand from background, and use histogram of oriented gradients (HOG) to extract hand features. Machine learning algorithms including *k*-nearest neighbor (*k*-NN), artificial neural network (ANN) and support vector machine (SVM) are used to classify hand gestures according to these hand features. Users can interact with virtual 3D objects through three common gesture manipulations, i.e. translational move, rotation, and zooming.

## 3D Display

In order to realize 3D interaction, it is necessary to build and render 3D objects. Unity is used as the 3D engine in which two virtual cameras spaced by inter-pupil distance (6.5 cm) are used to generate side-by-side 3D images. In addition, the optical see-through smart glasses must be calibrated so that the 3D virtual object is properly rendered. As shown in Fig. 1, user sees virtual object rendered by Unity through smart glasses in the Unity coordinate system. User’s hand position is tracked in the camera coordinate system and projected on the camera image plane. In order to achieve augmented reality interaction, user’s hand on camera coordinate system needs to be aligned with its counterpart in Unity.

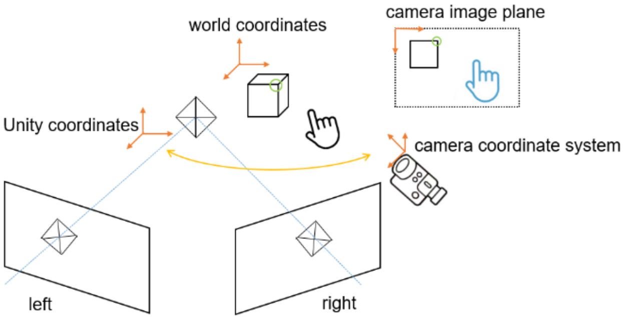


Fig. 1. Coordinate systems for Unity, camera and world.

The depth camera tracks the hand position $P_c(x_c, y_c, z_c)$ in the camera coordinate system, and projects it on camera image plane $P_s(u, v)$ with the depth $z_c$. The projection satisfies the equation below.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \qquad (1)$$

The position is represented by $P_c(x_c, y_c, z_c)$ in the camera coordinate system, and by $P_v(x_v, y_v, z_v)$ in the Unity coordinate system, respectively. These coordinates shall satisfy the relation as Equation 2. As the world coordinate system requires user’s head to be fixed or tracked, we transform camera coordinate system directly to Unity coordinate system for the sake of convenience.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KP_c = KT_{v\text{-}c}P_v = KT_{v\text{-}c} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} \qquad (2)$$

A set of 3D-2D point correspondences is needed to be collected as calibration data to solve the 3×4 transform matrix *KTv-c*. Each pair of point makes up 2 rows of equation, therefore, at least 6 pairs of point are needed to solve the matrix [7]. However, in order to reduce the measurement error, we collect more points and use least square estimation to solve the matrix. Then the transform equations are used to obtain the hand coordinates in the Unity coordinate system for augmented reality interaction.

### Hand Segmentation

In order to detect hand features, we need to segment the hand area from background in the captured image. Every pixel in the image is classified as hand pixel or background pixel according to the pixel's color information and depth information. Depth information is constructed by stereo camera, whereas color information is determined by the skin color model.

HSV (Hue, Saturation, and Value) color space is suitable for segmenting skin color region with good coverage for different human races. An empirical threshold value for HSV skin color model is shown in Equation 3.

$$\begin{aligned} &0 \le H \le 28 \\ &43 \le S \le 215 \\ &35 \le V \le 251 \\ &where\ H \in [0,180) \quad S,V \in [0,255] \end{aligned} \qquad (3)$$

Background that contains a color similar to skin can't be filtered by the skin color model. To get rid of the interference, depth information is introduced for hand segmentation. We choose the distance between 20 to 60 cm from the smart glasses as the interaction range, where user can interact with virtual objects by their hands. By the depth threshold hand image is separated from background. To achieve robust hand segmentation, both skin color and depth based segmentation are combined. As shown in Fig. 2, by combining the binary color image and depth image using bitwise AND operation, the hand segmentation is much improved. After hand segmentation, morphology filter operation is performed to remove image noise.

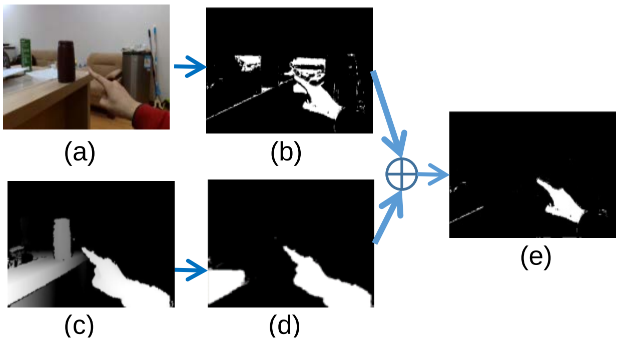


Fig. 2. Hand segmentation procedures: (a) capture color image, (b) filter by skin-color, (c) capture depth image, (d) threshold by depth, and (e) merge images

### Gesture Recognition

For robust and effective gesture recognition, the result of hand segmentation is adopted and features are extracted before machine learning and statistic gesture classification. The machine learning and gesture recognition flow chat is plotted in Fig. 3.

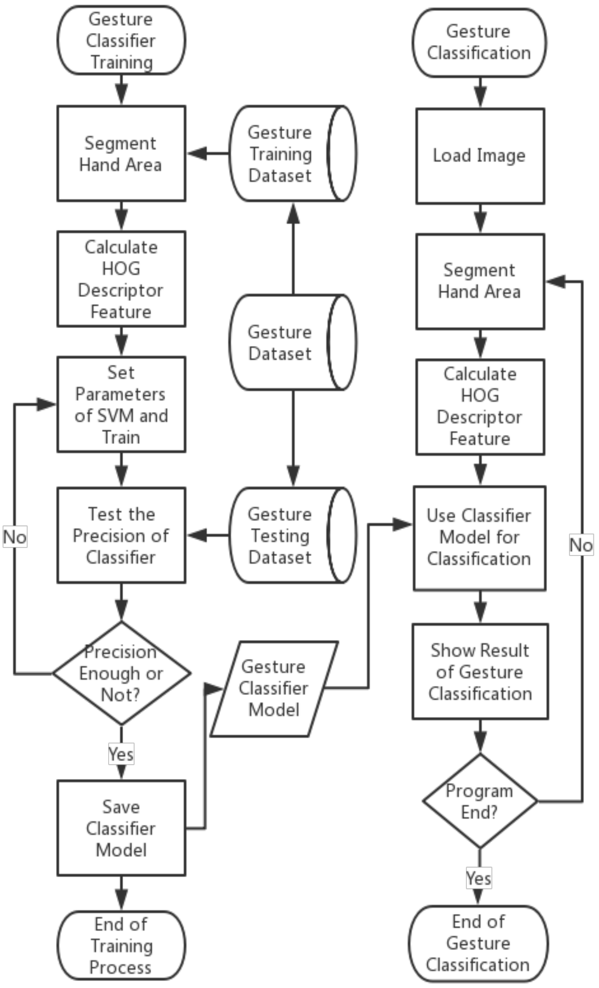

Gesture Classifier Training
Segment Hand Area
Gesture Training Dataset
Calculate HOG Descriptor Feature
Gesture Dataset
Set Parameters of SVM and Train
Test the Precision of Classifier
Gesture Testing Dataset
No
Precision Enough or Not?
Yes
Gesture Classifier Model
Save Classifier Model
End of Training Process
Gesture Classification
Load Image
Segment Hand Area
Calculate HOG Descriptor Feature
Use Classifier Model for Classification
No
Show Result of Gesture Classification
Program End?
Yes
End of Gesture Classification

Fig. 3. Flow chat of gesture recognition based on the machine learning.

Gesture recognition based on machine learning consists of training progress and gesture classification. The training progress uses supervised learning method which takes labeled training data as input to find the optimal parameters of classifier model. The optimal classifier model is saved during training progress and used for gesture classification. Without loss of generality, this paper adopts a gesture database [8] by Sebastien Marcel (Idiap Research Institute) for model training and testing.

### Hand features extraction

In order to classify hand gestures, hand features are extracted from hand area, which is segmented from color and depth images. This paper uses HOG descriptor [9] to represent hand shape feature. HOG descriptor describes the shape of the target object by a histogram of intensity gradient which is robust under various illumination. it is widely used on object detection in recent years and it's still a popular and suitable choice for hand pose estimation.

The segmented hand image is resized to 64×128 in grayscale before HOG feature extraction. The computation procedure of HOG is list as follows.

Algorithm 1. Computation of HOG features

1. Divide the image into overlapping blocks, and then each block is divided into several cells.
2. Calculate the gradient histogram of 9 directions for each cell, which forms a 9 dimensional feature vector.
3. Normalize the feature vectors for each block. Put the feature vectors together to get the HOG descriptor of the image.

In this paper, as shown in Fig. 4, the block size is 16×16 and each block contains four 8×8 cells. the block stride is set to 8, so the 64×128 hand image contains a total of 7×15 blocks.

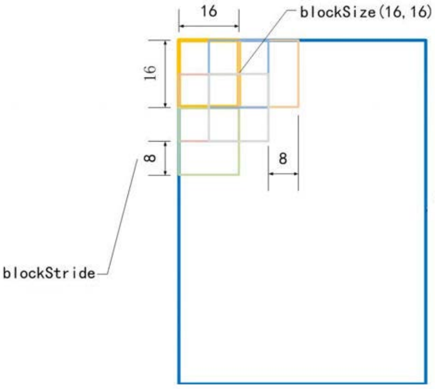


Fig. 4. Block size and stride

For each hand image, the dimension of HOG feature vector is 3780 as calculated in Equation 4, which is high enough for gesture classification.

$$dim = 9 \times M_{cell} \times N_{Block} = 9 \times 4 \times 7 \times 15 = 3780 \quad (4)$$

### Hand gesture classification

Hand gesture classification takes advantage of pre-trained classifier model to make prediction on hand image for classification. In this paper, three prominent machine learning algorithms, i.e. k-NN, ANN and SVM, are employed to make a comparison for hand gesture classification. This paper takes 5 gestures (fist, palm, bloom, point and pinch) out of The hand gesture dataset for 3D interaction. The 5 gestures in the dataset is shown in Fig. 5.

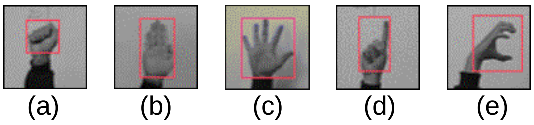


Fig. 5. Static hand gestures consist of (a) fist, (b) palm, (c) bloom, (d) point, and (e) pinch.

In machine learning, *k*-NN is one of the simplest algorithms for classification and regression problems. The algorithm simply compares a feature vector to be classified with k nearest feature vectors in the labeled train set, and assigns the vector the most frequent class. It's easy to implement due to its simplicity so it's used as a benchmark in algorithm comparison. The parameter k is chosen a moderate number k=11 in this paper.

Artificial neural network has been widely used in the area of computer vision and gesture recognition and been proved effective with good performance. A common architecture of feed-forward artificial neural network is multi-layer perception (MLP). The ANN architecture is composed of input layer, hidden layer and output layer as shown in Fig. 6.

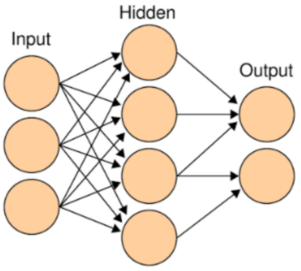


Fig. 6. Multi-layer perception

This paper uses a multi-layer perceptron network that has three layers for gesture classification. Input layer has 3780 neurons corresponding to the dimension of HOG feature vector. The number of neurons in hidden layer is 200 which is smaller than that of input layer and larger than that of output layer. Output layer has 5 neurons matching 5 gestures in the database. The used activation function is a hyperbolic tangent (tanh):

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (5)$$

Another machine learning algorithm this paper adopts is SVM [10]. SVM is a binary linear classifier which constructs a hyper plane as the decision surface to separate feature vectors into two class. It uses statistical learning theory to find the optimal separating hyperplane such that margin of between two separated class is maximized. The vector on the margin of optimal hyperplane is called support vector as shown in Fig. 7.

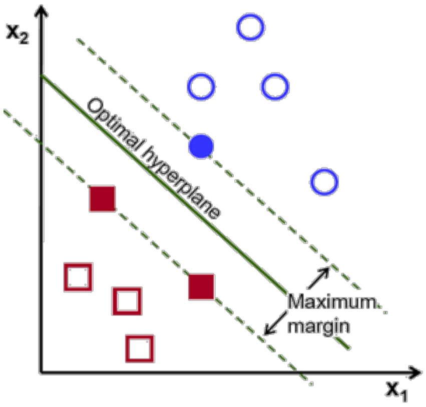


Fig. 7. SVM classification

SVM is a binary classifier which only separates two kinds of gesture. As for 5 hand gestures in database, this paper uses one-against-one method. A total of 5×(5-1)=20 binary classifiers are construct to classify the five hand gestures.

The training set of hand gesture dataset has 2835 hand gestures images while the test set has 547 hand images separated by uniform background (single color background) and complex background. The test results are summarized in Table 1.

Table 1. comparison of machine learning algorithms

| Machine learning algorithm | Uniform background | Complex background |
|---|---|---|
| k-NN | 43.8% | 38.6% |
| ANN | 86.7% | 61.3% |
| SVM | 90.9% | 72.6% |

As shown in Table 1, *k*-NN has low accuracy in both uniform background and complex background compared with ANN and SVM. The reason why SVM can achieve better accuracies is that the number of dimension (3980) of HOG feature is larger than the number (2835) of training examples.

Under complex lighting condition, the skin color model is unreliable, so the misclassification mainly results from the low resolution of depth camera. Using high-resolution depth camera and more training examples can improve the accuracy of gesture classification

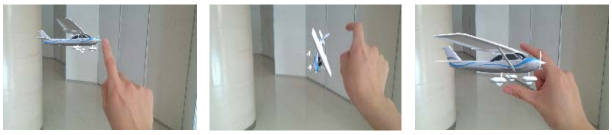
## Demonstration

A 3D gesture interaction system is built on EPSON BT-2000 smart glasses, which is equipped with a 5-million-pixel binocular depth camera. The hardware specifications of smart glasses are listed in Table 2.

Table 2. comparison of machine learning algorithms

| | |
|---|---|
| Field of view | 23 degrees (diagonal) |
| Platform | Android 4.0.4 |
| Main processor | OMAP4460 (Max 1.2 GHz) |
| Camera | 5 million pixels depth camera |

Our system creates virtual 3D objects using Unity 3D, and projects it to the lens of smart glasses. Users can interact with virtual objects by using hand gestures as shown in Fig. 7. Click the 3D object to select and move it. Slide index finger on horizontal or vertical axis to rotate the object. To zoom in or zoom out the object, use two fingers to pinch it. The distance accuracy of 3D hand interaction is within 2 cm.



(a) (b) (c)

Fig. 8. Gesture manipulations of (a) translation, (b) rotation, and (c) zooming.

## Acknowledgements

This work is sponsored by Shanghai Rockers Inc. (15H100000157).

## References

1. H. S. Yeo, B. G. Lee, and H. Lim, Multimed Tools Appl. vol. 74, p. 2687 (2015).
2. Y. Z., G. Jiang and Y. Lin, Pattern Recognition, vol. 49, p.102 (2016).
3. C. Qian, X. Sun, Y. Wei, X. Tang and J. Sun, IEEE Conference on Computer Vision and Pattern Recognition(CVPR) (2014).
4. T. Ha, S. Feiner, and W. Woo, International Symposium on Mixed and Augmented Reality (ISMAR) (IEEE, 2014).
5. C. Yu, W. Peng, S. Mao, Y. Wang, W. Chinthammit and H. B. Duh, Siggraph Asia, (2015).
6. K. Wang, C. P. Chen, L. Zhou, Y. Wu, B. Yu, and Y. Li, 1st International Conference on Display Technology (ICDT), Fuzhou (2017).
7. M. Tuceryan, Y. Genc, and N. Navab, Presence-Teleop. Virt. vol. 11, p. 259 (2002).
8. A. Just, Y. Rodriguez, and S. Marcel, International Conference on Automatic Face and Gesture Recognition (FG), (IEEE, 2006).
9. N. Dalal and B. Triggs, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2005).
10. C. Cortes and V. Vapnik, Mach. Learn. vol. 20, p. 273 (2015).