REAL-TIME HAND GESTURE RECOGNITION BASED ON SUPPORT VECTOR MACHINE WITH DEPTH SENSOR

[1] *Cheng-Hao Wang* (*王政皓*), [2] *Chiou-Shann Fuh* (*傅楸善*)

[1] Dept. of Bio-Industrial Mechatronic Engineering,
National Taiwan University, Taiwan
r03631013@ntu.edu.tw

[2] Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

ABSTRACT

In this article, we present a real-time hand gesture recognition system that can successfully recognize a number of gestures. Our system consist of two major part, hand detection and gesture recognition. We use Kinect sensor to capture both color and depth information, then segment out hand region with skin color model and proper distance. Then, in gesture recognition, we first try to obtain the shape representation with the contour of the hand, which is called time-series curves. The principal components analysis (PCA) is then used to reduce the dimensionality of the features of time-series curves. With the reduced features, SVM is trained to recognize the hand gestures, and the mean accuracy can achieve 93.675% with 0.208 mean running time.

Keywords *Hand Gesture Recognition, Kinect Sensor, Skin color detection, PCA, Support Vector Machine (SVM).*

1. INTRODUCTION

In recent years, touch panel technology has been well received by the users. However, it still has a limitation of distance during operation, and mostly controlled by the fingertips. Hand gestures recognition is growing more importance in human-computer interaction (HCI), it contributes to a wide range of application in virtual reality, sign language recognition, and computer games.
However, vision based hand gesture recognition is a challenging problem because of variant lighting condition and complicate background, which will make the detection of hand very unstable and low-quality.

To enable a more robust hand gesture recognition, Zhou Ren and Junsong Yuan [1] used Kinect depth sensor instead of optical sensor to overcome the problems lighting condition and complicate background. Then they segment out the hand region and obtain shape representation of hand by recording the distance between hand contour point and center point, which is called time-series curve containing the information of fingers. And they present a novel shape distance metric called Finger-Earth Mover's Distance (FEMD) for hand gesture recognition with approximate 90% mean accuracy.

Nasser H. Dardas and Nicolas D. Georganas[2] use Scale Invariant Feature Transform (SIFT) to perform feature extraction, then apply bag-of-features approach to acquire feature vectors of fixed and reduced dimension of key points. After features extraction, feature vectors of different hand gestures are used to build a muti-class SVM model for gesture recognition. The result showed that, the recognition accuracy can be approximate 95% and recognition time about 0.017 second which present a fast and robust result.

Chen and Tseng [3] simply set three webcams to capture different angles of images of hand, then resize the images to smaller image and convert to grayscale that retains enough details of gestures. For gesture recognition, multi-class SVM classifier was build with three different gestures, and the overall accuracy can be 73.3%.

In our approach, we used Kinect sensor to perform robust hand segmentation, and obtain hand shape with time-series curves. Then we use PCA method to extract out major features of the time-series curve to form feature vectors. Finally, these feature vectors of different hand gestures are fed to multiclass SVM classifier for hand gesture recognition.

2. HAND DETECTION

2.1 Hade Detection Based On Skin Color

In the process of skin color detection, image is very sensitive to the brightness of ambient light. If we detect the skin color based on RGB color space, it will be very unstable and cause wrong detection. As a result, transformation of the image from RGB to other suitable color space is needed. YCbCr color space is known a better choice because it can separate out the brightness element Y. Using only the information of Cb(blueness) and Cr(redness) which are related to the hue can largely reduce the effect of light change, and reach more stable result. The transformation from RGB to YCbCr :

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{Cb} \\ \mathrm{Cr} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.31126 & 0.5 \\ 0.5 & -0.51869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (1.1)$$

Fixed color model result still has some problem because of the ambient change of light and the skin color variance between different operators. Therefore, we came up of an idea that grab the partial hand image of operator covering the palm as large as possible every time before operation, and transform it to YCbCr, then build the color model based on mean value and standard deviation of Cb, Cr. x_i is the Cb, Cr value of pixels of hand, N is the number of pixels:

$$m = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (1.2)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - m)^2} \qquad (1.3)$$

Then we can create the skin color model of hands in range of two standard deviation from the mean values:

$$\text{if}\begin{cases} m_{Cb} - 2\sigma_{Cb} < Cb(x,y) < m_{Cb} + 2\sigma_{Cb} \\ m_{Cr} - 2\sigma_{Cr} < Cr(x,y) < m_{Cr} + 2\sigma_{Cr} \end{cases}, Img(x,y) = 1$$
$$else, Img(x,y) = 0 \qquad (1.4)$$

Besides, those pixels nearby the already defined skin pixels have high probability of being skin pixels, so we can enlarge the range:

$$\text{if } (\mathrm{Img}(x-1,y-1) = 1)\mathrm{OR}(\mathrm{Img}(x-1,y) = 1)\mathrm{OR}(\mathrm{Img}(x,y-1) = 1$$
$$\text{AND if } \begin{cases} m_{Cb} - 4\sigma_{Cb} < Cb(x,y) < m_{Cb} + 4\sigma_{Cb} \\ m_{Cr} - 4\sigma_{Cr} < Cr(x,y) < m_{Cr} + 4\sigma_{Cr} \end{cases}, \qquad (1.5)$$
$$\mathrm{Img}(x,y) = 1$$

With this skin color model, we can segment out the skin colored objects in a cluttered environment as Fig. 1.

Figure 1. : (a) Experimental surrounding (b): Skin colored object

2.2 Hand detection based on depth image

Using Kinect sensor can get the depth information of the image. With the depth information, targeting on a proper distance can segment out the image of hand. Therefore, we first set [5, 30] as the working distance of the hand, so the pixels out of this range are set to 0s, and pixels in this range are set to 255s. Then we try to find the minimum distance r_{min} in this range, and we target on a more precise range relative to the minimum distance [$r_{min} - 2, r_{min} + 2$], a rough hand region can be obtained in Fig. 2. For this implementation, we require the user to make sure that the hand is the frontmost object facing the sensor.

However, because Kinect sensor has sever problem of noise and shadow, as you can see the broken parts near the edge of the hand in Fig. 2. Applying median filter can relieve this problem, then using morphological close operation to further fill the broken parts and obtain more complete hand image as Fig. 3 shows.

(a) (b)

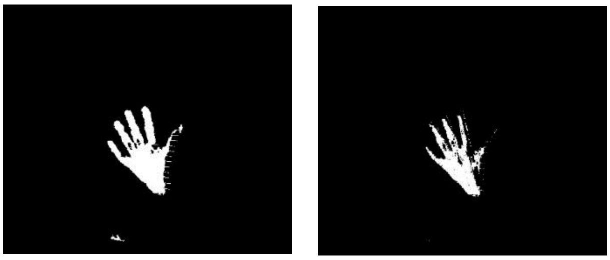


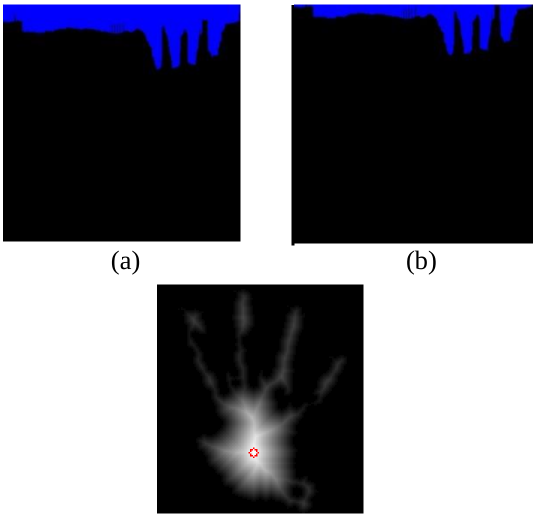
Figure 2. : (a) (b): Hand region segmented out by using depth image only and both depth image & skin color image. You can see from (a) that there is something in the same range as hand, so applying skin detection can avoid this problem.

(a) (b)

Figure 3. : (a) (b): Hand image before median filtering & close operation and after it.

2.3 Hand Gesture Representation

After detecting the hand region, we try to represent the shape of the hand. The representation measures the relative distance between the contour points and a center point, which is called time-series curves. The center point is defined as the point with maximal distance relative to the contour using Distance transform on the hand region, as shown in Fig. 4(c). Then we find the angle and Euclidean distance between each contour point and center point, and denote the angles on the horizontal axis and the distance on the vertical axis. As shown in Fig. 4(a), the curve has good representation of the shape of hand gesture, such that the fingers appear local maximal distance from the center point. However, we need only the finger clusters in the time-series curves, so we detect the minimum distance on vertical axis, and treat it as an approximate distance from the center point to the beginning of the finger. Subtracting the minimum distance from the heights of time-series curves, we can obtain a new time-series curve that emphasize the information of fingers more as shown in Fig. 4(b).



(a) (b)

(c)

Figure 4. : (a) (b): Time-series curve images before and after subtracting the minimum distance, as can be seen that the clusters of fingers are emphasized and redundant information is removed. (c): The center point (red point) obtained from distance transform image.

3. HAND GESTURE RECOGNITION

Our hand gesture recognition system consists of two major part: feature extraction from time-series curves of hand gestures and SVM classifier construction of gestures of number one to four. Then the classifier model will be used to recognize hand gestures obtained from Kinect sensor.

3.1 Feature Extraction of Hand Gestures

First, we resize the time-series curve images to 60*60 to reduce the features and workload of classifier training. The new image size still retain enough information of gestures. The time-series curve images were acquired from five different hand gestures, and to adapt to the rotation of hands, we randomly rotate hands with suitable angles that is normal to the poses of human hand. So the hand gesture recognition system will be more adaptive to a slight rotation of the hand.

Then, to extract out the major features, the PCA method was used to reduce the dimensionality of the time-series curve images. We concatenate each time-series curve image by rows that each sample image forms a 1×3600 vector, and use PCA method to extract out the major features.

We give a brief introduction of Principle component analysis (PCA) as below :

The PCA method is a popular dimensionality reduction technique to find a set of orthonormal vectors in the data space, which can maximize the data's variance and map the data onto a lower dimensional subspace spanned by those vectors. Consider a data with L images $x_i \epsilon R^N$ $(i = 1, \cdots, L)$, and N is the number pixels in the image. The total scatter matrix $S_T \in R^{N \times N}$ is defined as

$$S_T = \sum_{i=1}^{L} (x_i - \mu)(x_i - \mu)^T = AA^T \qquad (2)$$

$A = [x_1 - \mu \cdots x_L - \mu] \in R^{N \times L}$. Where μ is the global mean image of the training set, and

A computation of S_T is impractical due to the huge size $N \times N$ of the matrix. Instead of direct finding the eigenvector W_{PCA} of S_T , we solve the eigenvalue problem, $RV_{PCA} = V_{PCA}\Lambda$, to obtain the eigenvectors, $V_{PCA} \in R^{L \times P}$, and the eigenvalues, $\Lambda = \mathrm{diag}[\lambda_1 \cdots \lambda_P] \epsilon R^{P \times P}$, with decreasing order $\lambda_1 \geq \cdots \geq \lambda_P > 0$, where λ_i is the nonzero eigenvalues of the matrix $R = A^T A \in R^{L \times L}$. Then, the PCA subspace W_{PCA} is formed by multiplying the matrix A with the eigenvector that is, $W_{PCA} = AV_{PCA} \in R^{N \times P}$. Therefore, the feature vector **y** of an image **x** is acquired by projecting **x** into the coordinate system defined by the PCA subspace, that is

$$y = W_{PCA}^T (x - \mu) \epsilon R^P \qquad (3)$$

3.2 SVM Classifier Construction

3.2.1 Support Vector Machines

SVM is a classification method that aims to separate two data sets with maximum distance between them. It is proposed by Vapnik and his co-workers [9]. This method separates two data sets by searching for an optimal separating hyperplane between them (Fig. 5). Bounds between data sets and optimal separating hyperplane are called "support vector".

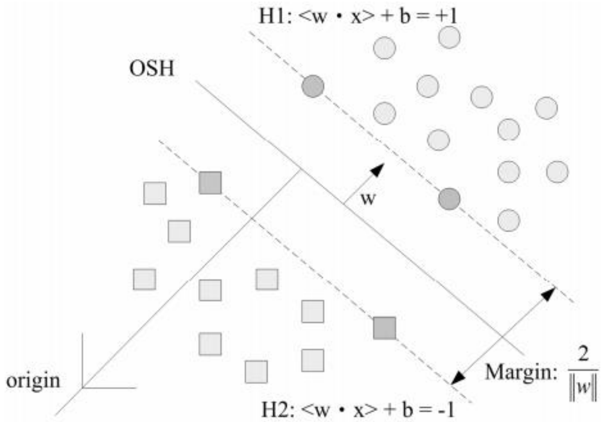


Figure 5. Optimal separating hyperplane

Each point in total data set is referred as $x_i \in IR^n, i = 1,2, \ldots, N$ and belongs to a class $y_i \in \{-1,1\}$. For linear classification we can identify two classes and the optimal separating hyperplane separating them like,

$$\mathrm{w} \cdot x_i + b \geq 1, \ y_i = 1 \qquad (4)$$

$$w \cdot x_i + b \geq 1, \quad y_i = 1 \tag{5}$$

We can generalize (4) and (5) with the form.

$$y_i \cdot [(w \cdot x_i) + b] \geq 1, \quad i = 1, \ldots, l \tag{6}$$

The distance between support vectors are pre-defined as:

$$d = \frac{2}{\|w\|} \tag{7}$$

The bigger d is, a better separation between two classes can be achieved. For this reason to maximize d we need to minimize norm of w. This problem can be solved using Lagrange function.

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^{l} \alpha_i \cdot \{y_i \cdot [(w \cdot x_i) + b] - 1\} \tag{8}$$

Here α_i represents Langrange multipliers. Solving (8) by minimizing according to w and b, maximizing according $\alpha_i \geq 0$ values, most suitable OSH parameter w can be obtained in (9) according to condition $\sum_{i=1}^{l} \alpha_i \cdot y_i = 0, \alpha_i \geq 0, i = 1, \ldots, l,$

$$w = \sum_{i=1}^{l} \alpha_i \cdot y_i \cdot x_i, \alpha_i \geq 0, i = 1, \ldots, l \tag{9}$$

Distance of any data point x to OSH is defined as :

$$d(w, b; x) = \frac{|w \cdot x + b|}{\|w\|} \tag{10}$$

We can get a more generalized form of (10) by replacing w with its value shown in (9),

$$d(x) = \frac{\left(\sum_{i=1}^{l} \alpha_i \cdot y_i \cdot x_i\right) \cdot x + b}{\left\|\sum_{i=1}^{l} \alpha_i \cdot y_i \cdot x_i\right\|} \tag{11}$$

Sign of distance calculated in (11) shows us to which class point x belongs and $|d|$ shows distance of x to OSH. As $|d|$ increase a better classification result can be obtained.

Linear separation of data sets can not be achieved successfully all the time. In such cases a simple conversion of feature space is done. Point in first data space is expanded to a feature space with higher dimension and linear separation is retires. This expansion process is realized with operator $\Phi(\cdot)$ OSH function turns into the form:

$$f(x) = w \cdot \phi(x) + b \tag{12}$$

By replacing w with its value in (9) we can get a more generalized form as:

$$f(x) = \sum_{i=1}^{l} \alpha_i \cdot y_i \cdot \left(\phi(x_i) \cdot \phi(x)\right) + b \tag{13}$$

In a high dimensional space realization of $\cdot \left(\phi(x_i) \cdot \phi(x)\right)$ multiplication is intractable. For this reason "Kernel Function" in $K(x_i, x_j) = \left(\phi(x_i) \cdot \phi(x)\right)$ form are used. In such processes there are two widely used kernel functions:

(i) Polynomial Kernel Function

$$K(x_i, x_j) = (x_i \cdot x{+}1)^p \tag{14}$$

(ii) RBF Kernel Function:

$$K(x_i, x_j) = \exp[-\gamma \|x - x_i\|^2] \tag{15}$$

We examined classification of only RBF kernel function in this study.

SVM has been successful in application of image recognition such as in [2]. Though SVM is initially intended as binary classifier, other methods can handle multiclass problem by creating all pair of two-class classifiers.

3.2.2 Training stage

In training stage, the sample images is collected from four hand gestures representing number one to four, then resize to 60*60 as Figure 6 shows. 2000 resized time-series curve images are collected for each gesture, then reshape each image to form a vector of size 1*3600, and perform PCA method to extract out the major features of the time-series curve images to obtain a feature vector, and use it for training. After feature extraction, we fed every feature vector with its related label of numbers 1~5 into SVM classifier to construct the multiclass SVM classifier. The training stage is shown in Fig. 7.

Figure 6. Resized 60*60 hand resized time-series curve images.

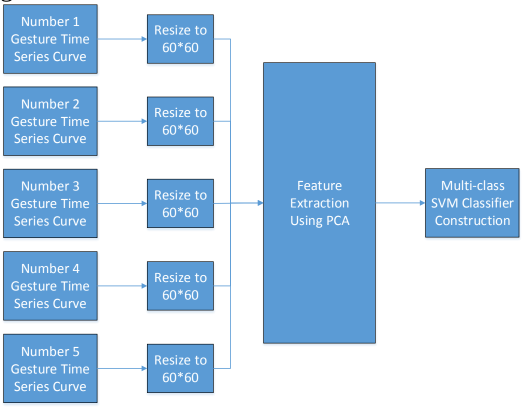


Figure 7. The training stage for multi-class SVM classifier.

4. EXPERIMENT RESULT

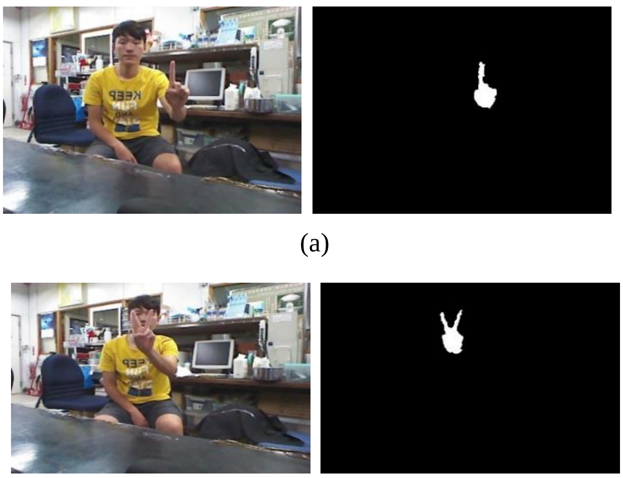
4.1 Performance evaluation

Our hand gesture recognition system is robust to cluttered backgrounds, since the hand shape is detected using both depth and color information, therefore the background can be effectively removed. Fig. 8(a) illustrate an example when the hand is cluttered by the background, which is difficult for other hand gesture recognition methods using color information only. Fig.

8(b) shows another case that will be extremely difficult for the skin color-based hand gesture recognition approaches when the hand is confused with the user's face. Whereas our hand segmentation can be very accurate using Kinect sensor.

In testing stage, we tested four hand gestures that represent number 1 to 5 by video files captured from Kinect sensor. Fig. 9 shows the four hand gestures captured and the corresponding hand contour and time series curves. Each video with 100 frames were transformed to time series curves of hand, then 50 features extracted by PCA were tested by multi-class SVM classifier to evaluate the performance. Table 1 shows the accuracy and the mean running time of multi-class SVM classification for different hand gestures. The mean accuracy of time series curve based multi-class SVM is 95.4%, and the mean running time is 0.163s, which is a particularly suitable method for real-time gesture recognition. Table 2. shows the performance of the method we presented comparing to the other methods that have been proposed before. Our method is superior to the traditional correspondence-based matching algorithm, shape context[3] in accuracy and time, but inferior to the others in time. Despite of this, we have other advantages such as the operator doesn't need to wear anything and it can be operated in a cluttered background without confusion with other skin-color objects. Fig. 10. Shows some correct samples for hand gesture recognition using multiclass SVM classifier.

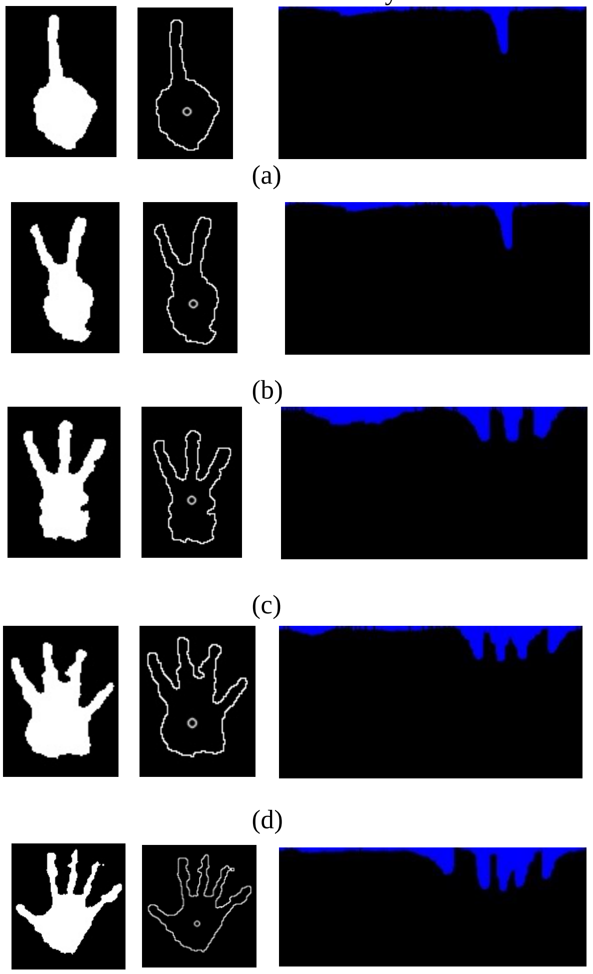
Fig. 11. Shows the confusion matrix of gesture recognition. Number 2 and 3 gesture have the highest accuracy achieving 98%, number 3 and number 4 gesture have good accuracy near 95%, and the accuracy of number 5 gesture is apparently low. It means that the performance between gestures can be quite different, and may be affected by the feature extraction method and feature number fed in to the multiclass SVM classifier. We can achieve a better result through improving the feature extraction method or selecting gestures that have relatively good performance.



(a)

(b)

Figure 8. Our hand segmentation result is robust to the cluttered background. (a) The hand in the cluttered background is detected accurately; (b) The hand in front of the face can be detected accurately.



(a)

(b)

(c)

(d)

(e)

Figure 9. Four hand gestures representing number 1 to 4 are detected and transformed to time series curves in the right column.

4.2 Performance of Different number of Features

Logically speaking, the more features are acquired from the PCA method will better include complete information of hand gesture. However more features doesn't means absolutely higher recognition accuracy, because redundant or irrelevant information may be wrongly included in the process of SVM classifier construction, and cause lower accuracy. Besides, too many features may lead to over-fitting as well.

For evaluating the performance of different number of features, we used PCA to extract other four different number of features－100, 200, 500 and 1000, then fed into the SVM and test for recognition performance.

Table 1. and Table 3. to Table 6. show the results of different number of features. The results show that number 1 and number 2 gesture possess the highest recognition accuracy, number 3 and number 4 gesture

possess the second high accuracy which is not much lower than number 1 and number 2, and the accuracy of number 5 gesture falls abruptly except for the case of 50 features.

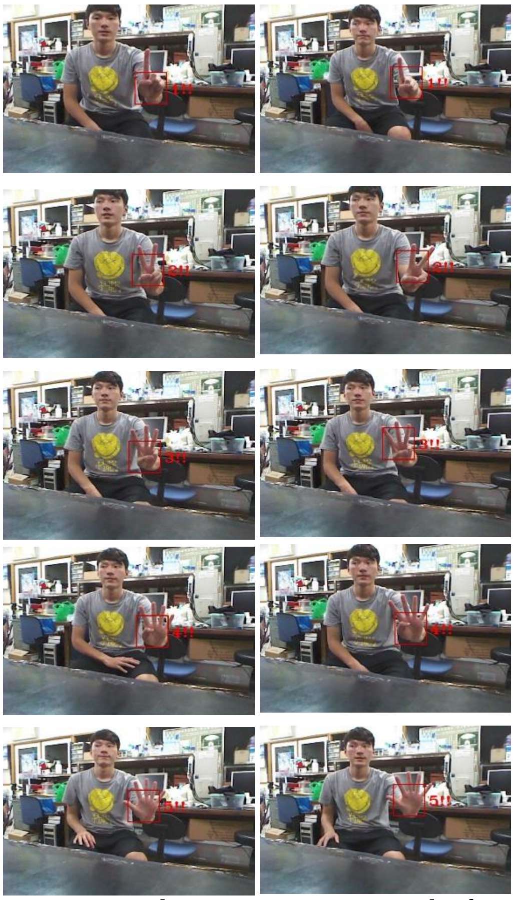


Figure 10. Hand gesture recognition result of number 1 to 5 gestures.

On the other hand, we compare the results between different number of features. It shows that the recognition accuracy bec-omes lower and recognition time becomes longer as the more features are included. This may be due to the irre-levant features which are also included as we have mentioned above. Additionally, the result greatly demonstrates the effectiveness of PCA method used to extract the major features of hand gestures and reduce the amount of workload. Figure 13. and Figure 14. show the results of recognition accuracy and recognition time versus different number of features.

Table 1. Performance of hand gesture recognition based on multi-class SVM classification with 50 features.

Gesture	Number of frames	Correct	accuracy	Running Time(s)
One	300	294	98%	0.161
Two	300	294	98%	0.162
Three	300	285	95%	0.163
Four	300	286	95.3%	0.160
Five	300	273	91%	0.168
Mean	300	286.4	95.4%	0.163

1	0.98				0.02
2	0.02	0.98	0.03		0.01
3		0.02	0.95	0.05	
4			0.01	0.95	0.06
5			0.01		0.91
	1	2	3	4	5

Figure 11. The confusion matrix of gesture recognition using multi-class SVM classification.

4.3 Robustness to slight rotation and scale change

In practical operation, hand gestures will not remain the same angle, and scale will not remain the same either. As the position of the hand changes or the variance of gesture pose between different operators, there would be slight rotation and scale change of the hand gesture. To adapt to this situation, we have trained SVM with sample images that exist slight angle and scale changes. Fig 12. shows the gesture recognition result of slight rotation and scale change. It appears that our method can be robust to the change in a proper range of variance. However, because of the working distance that is predefined in a range of [5,30] in depth value of Kinect, the change in scale is limited, and the accuracy will decrease as the change become more, since some information are lost. For example, the length of finger clusters decrease, so the information of the finger region are reduced. Besides, too much rotation will lead to lower accuracy too. Because the time-series curve will change significantly, and make the finger region appear in wrong places.

However, we can utilize the method of normalization to solve this problem.

6. CONCLUSION

In this paper, we have presented a real time hand gesture recognition system consist of three part: hand detection based on color and depth image, gesture shape representation using time-series curve, and gesture recognition using PCA method for feature extraction and multi-class SVM classification. Experiment result shows high accuracy and applicable short running time operated

Table 2. The mean accuracy and the mean running time of our method comparing to the other method that have been proposed.

	Mean Accuracy	Average Running Time(s)
Thresholding Decomposition+FEMD[1]	93.2%	0.0750
Bag-of-Features+SVM[2]	96.23%	0.017
Shape Context without bending cost[4]	83.2%	12.346
Time Series Curve + PCA + SVM	95.4%	0.163

in cluttered environment, and can be adaptive to a little rotation. However, the performance varies in different gestures. It means that the feature extraction method can be adjusted to include more dominant features as possible and still remain short running time.

7. REFERENCES

[1] Ren, Zhou, Junsong Yuan, and Zhengyou Zhang. "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera." *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011.

[2] Ren, Zhou, et al. "Robust hand gesture recognition with kinect sensor."*Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011.

[3] Dardas, Nasser H., and Nicolas D. Georganas. "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques." *Instrumentation and Measurement, IEEE Transactions on* 60.11 (2011): 3592-3607.

[4] Chen, Yen-Ting, and Kuo-Tsung Tseng. "Multiple-angle hand gesture recognition by fusing SVM classifiers." *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*. IEEE, 2007.

[5] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, pp. 509–522, 2002.

[6] Huang, Deng-Yuan, Wu-Chih Hu, and Sung-Hsiang Chang. "Vision-based hand gesture recognition using PCA+ Gabor filters and SVM." *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on*. IEEE, 2009.

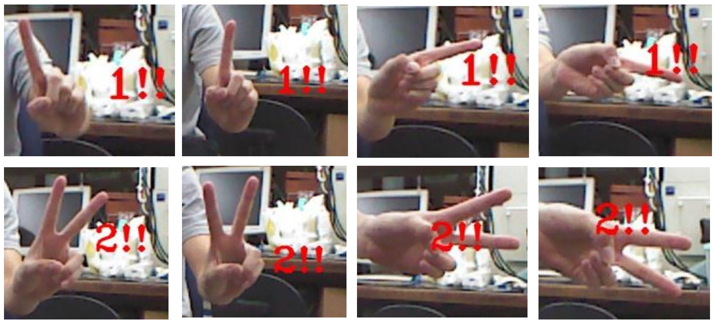
[7] Raheja, Jagdish L., Ankit Chaudhary, and Kunal Singal. "Tracking of fingertips and centers of palm using kinect." *Computational intelligence, modelling and simulation (CIMSiM), 2011 third international conference on*. IEEE, 2011.

[8] Phung, Son Lam, Abdesselam Bouzerdoum, and Douglas Chai. "Skin segmentation using color pixel classification: analysis and comparison." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.1 (2005): 148-154.

[9] Vapnik, V. "The nature of statical learning theory." (1995).

[10] Wang, Youwen, et al. "Kinect based dynamic hand gesture recognition algorithm research." *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*. Vol. 1. IEEE, 2012.

[11] Ren, Yu, and Fengming Zhang. "Hand gesture recognition based on MEB-SVM." *Embedded Software and Systems, 2009. ICESS'09. International Conference on*. IEEE, 2009.



(a)

(b)

Fig 12. Hand gesture recognition result against. (a)rotation. (b)scale.

[12] Ren, Zhou, et al. "Robust hand gesture recognition with kinect sensor."*Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011.

[13] Kurakin, Alexey, Zhengyou Zhang, and Zicheng Liu. "A real time system for dynamic hand gesture recognition with a depth sensor." *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012.

[14] Li, Yi. "Hand gesture recognition using Kinect." *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*. IEEE, 2012.

[15] Doliotis, Paul, et al. "Comparing gesture recognition accuracy using color and depth information." *Proceedings of the 4th international conference on PErvasive technologies related to assistive environments*. ACM, 2011.

[16] Zafrulla, Zahoor, et al. "American sign language recognition with the kinect."*Proceedings of the 13th international conference on multimodal interfaces*. ACM, 2011.

[17] Yao, Yuan, and Yun Fu. "Contour model-based hand-gesture recognition using the Kinect sensor." *Circuits and Systems for Video Technology, IEEE Transactions on* 24.11 (2014): 1935-1944.

[18] Keskin, Cem, et al. "Real time hand pose estimation using depth sensors."*Consumer Depth Cameras for Computer Vision*. Springer London, 2013. 119-137.

Table 3. Performance of hand gesture recognition based on multi-class SVM classification with 100 features.

Gesture	Number of frames	Correct	accuracy	Running Time(s)
One	300	295	98%	0.160
Two	300	285	95%	0.165
Three	300	279	92%	0.161
Four	300	287	96%	0.164
Five	300	215	72%	0.167
Mean	300	272.2	90.8%	0.163

Table 4. Performance of hand gesture recognition based on multi-class SVM classification with 200 features.

Gesture	Number of frames	Correct	accuracy	Running Time(s)
One	300	291	97%	0.168
Two	300	297	99%	0.168
Three	300	279	93%	0.171
Four	300	288	96%	0.173
Five	300	192	64%	0.172
Mean	300	269.4	86.8%	0.170

Table 5. Performance of hand gesture recognition based on multi-class SVM classification with 500 features.

Gesture	Number of frames	Correct	accuracy	Running Time(s)
One	300	296	99%	0.182
Two	300	290	97%	0.182
Three	300	271	90%	0.187
Four	300	236	79%	0.188
Five	300	174	58%	0.186
Mean	300	253.4	84.6%	0.185

Table 6. Performance of hand gesture recognition based on multi-class SVM classification with 1000 features.

Gesture	Number of frames	Correct	accuracy	Running Time(s)
One	300	289	96.3%	0.207
Two	300	288	96%	0.206
Three	300	276	92%	0.207
Four	300	251	84%	0.212
Five	300	160	53%	0.213
Mean	300	252.8	84.3%	0.209

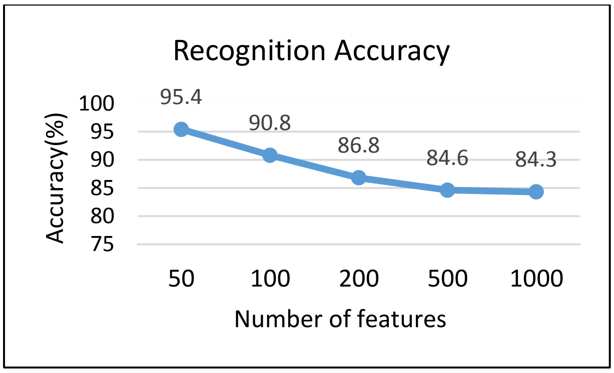


Figure 14. Results of recognition accuracy versus different number of features.

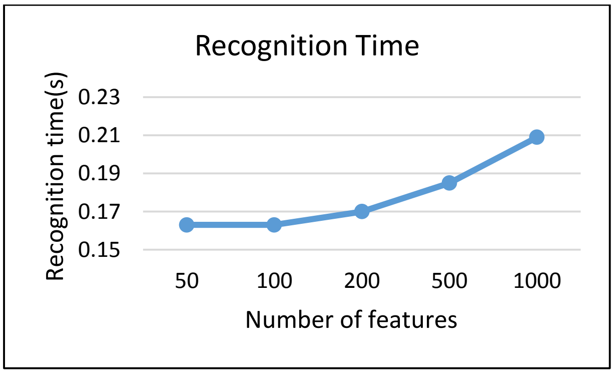


Figure 15. Results of recognition time versus different number of features.