Real-Time Hand Gesture Recognition Based on the Depth Map for Human Robot Interaction

Minoo Hamissi[1], Karim Faez[2]
[1]Department of Electrical and Computer Engineering, Islamic Azad University of Qazvin
[2]Department of Electrical Engineering, Amirkabir University of Technology

Article Info

Article history:

Received Jul 19, 2013
Revised Oct 2, 2013
Accepted Oct 26, 2013

Keyword:

Depth map
Hand gesture recognition
Human robot
Vocabulary tree

ABSTRACT

In this paper, we propose and implement a novel and real-time method for recognizing hand gestures using depth map. The depth map contains information relating to the distance of objects from a viewpoint. Microsoft's Kinect sensor is used as input device to capture both the color image and its corresponding depth map. We first detect bare hand in cluttered background using distinct gray-level of the hand which is located near to the sensor. Then, scale invariance feature transform (SIFT) algorithm is used to extract feature vectors. Lastly, vocabulary tree along with K-means clustering method are used to partition the hand postures to ten simple sets as: "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" and "ten" numbers based on the number of extended fingers. The vocabulary tree allows a larger and more discriminatory vocabulary to be used efficiently. Consequently, it leads to an improvement in accuracy of the clustering. The experimental results show superiority of the proposed method over other available approaches. With this approach, we are able to recognize 'numbers' gestures with over 90% accuracy.

Copyright © 2013 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Minoo Hamissi
Department of Electrical and Computer Engineering, Islamic Azad University of Qazvin
Qazvin Islamic Azad University - nokhbegan Blvd. qazvin. iran
(+98-281)3665275-3665276-3665277
Email: hamissi.minoo@gmail.com

1. INTRODUCTION

Nonverbal communication can be efficiently used for sending and receiving messages between people. Gestures and touch, body language or posture, physical distance, facial expression and eye contact are all types of nonverbal communication. Hand gestures provide suitable and efficient interface between human and computer. Particularly, using hand gestures, simple commands can be transmitted to the computer or personal robots in the real time. In fact, hand gestures are mainly developed for wordless or visual communication in human computer interfaces (HCIs). However, the recognition of hand gestures is very challenging in environment with cluttered background and variable illumination. Furthermore, real-time performance and recognition accuracy which are two requirements of the HCIs have to be considered in this field. Up to now several gesture recognition techniques have been proposed to meet these requirements. Early systems usually require markers or colored gloves to make the task easier such as [1], which use a dark glove with color coded ring markers to detect the fingertips. Nevertheless, using of markers and gloves has some limitations for the user's convenience.

Hand gesture recognition systems can be categorized into two classes [2]: the 3-D hand model-based methods, and the appearance-based methods [3].

The 3-D hand model-based technique [4-7] compares the input frames and the 2-D appearance projected by the 3-D hand model. These methods have high degrees of freedom and are based on 3-D kinematic hand model. This class has two major drawbacks. The 3-D hand models provide wide class of hand gestures. Consequently, a huge image database is required to deal with the entire characteristic shapes under several views. Second problem is the difficulty of feature extraction and inability to handle singularities which occur from unclear views.

The appearance-based techniques which use 2-D image features have attracted extensive interest so far. The real-time performance is the main advantage of this class. In these schemes, image features are extracted to model the hand. Then, these features are compared with the video frames.

One study [8] reported a method based on color of skin in the image. But, this method is very sensitive to lighting conditions and required that no other skin-like object exist in the image. In [9], hand postures are represented in terms of hierarchies of multi-scale color image features at different scales, with qualitative inter-relations in terms of scale, position and orientation. Although, the proposed algorithm shows real-time performance, it cannot recognize hand gesture in the image where other skin-colored objects exist. In another work, Argyros et al. [10] introduced an algorithm for controlling a computer mouse via 2D and 3D hand gestures. This method is vulnerable against noise and variable illumination in the clutter background. Some researchers focused on local invariant features [11-13].

In [11], using Adaboost learning algorithm and SIFT features leads to the rotation invariant hand detection. SIFT method [14] is a robust feature detection to represent image based on key-points. The key-points provide rich local information of an image. However, several features such as a contrast context histogram had to be used to achieve hand gesture recognition in real time.

In order to achieve real-time performance and high recognition accuracy, Haar-like features and the AdaBoost learning algorithm were suggested by Chen et al. in [13]. Juan et al. [15] evaluated performance of SIFT, principal component analysis (PCA) – SIFT, and speeded up robust features (SURF) by many experiments. SIFT algorithm extracts features which is invariant to the rotation and scale from images. PCA-SIFT, which is introduced in [16], employs PCA to normalize gradient patch. In [17], robust SURF features are used for Fast-Hessian detector and image convolutions.

Here, we focus on bare hand gesture recognition without help of any markers and gloves. To be robust against cluttered background and various lighting conditions, we used depth map which contains information relating to the distance of objects from a viewpoint. For this aim, Kinect sensor is utilized to capture both the color image and its corresponding depth map. Using the depth map, hand can be accurately detected according to distinct gray-level in our test environment. The detected hand is extracted by replacing hand area with a black circle. After extracting the hand, the hand area only is saved in a small image, which will be used in extracting the features by scale invariance feature transform (SIFT) algorithm. For the first time, Lowe [14] proposed using SIFT features which are invariant to scale, orientation and partially invariant to illumination changes, and are extremely distinctive of the image. Therefore, SIFT features are extracted from the hand detected images. After this step, a vocabulary tree is offline trained by the hierarchical K-means clustering. Next, a weighted vocabulary tree using Term Frequency Inverse Document Frequency (TFIDF) weighting is build to recognize numbers using k-nearest neighbor and voting. Figure 1 shows an overall picture of system.

The remainder of the paper is organized as follows. In section 2, hand detection and SIFT algorithm for feature extraction are explained. Section 3 describes offline training of the vocabulary tree using K-means clustering. Experimental results and comparison to the other state of the art methods are discussed in section 4. Finally, section 5 concludes the paper [4].

2. HAND DETECTION AND FEATURE EXTRACTION USING SIFT ALGORITHM

2.1. Hand Detection

Having a reliable hand detector in the clutter background and various lighting conditions is the main requirement of our system. The Kinect 3-D camera [18], with its depth sensing capability, provides the depth image in 640×480 resolutions at 30 fps. The depth information which is captured by an infrared camera will be converted into a gray scale image. Figure 2 shows an original depth map. In order to accurately extract the hands by judging the depth, the person's hands have to be in the front. Owing to low color contrast in the raw depth image, gray level rescaling is required. As a result, by adjusting the scale factor, we make the body and the background invisible. Besides, the hand is in gray scale and visible. Therefore, the hand, the gray part, can be extracted from the depth map by thresholding the gray level. Finally, we extracted a hand shape image for recognition. This procedure is depicted in Figure 3.

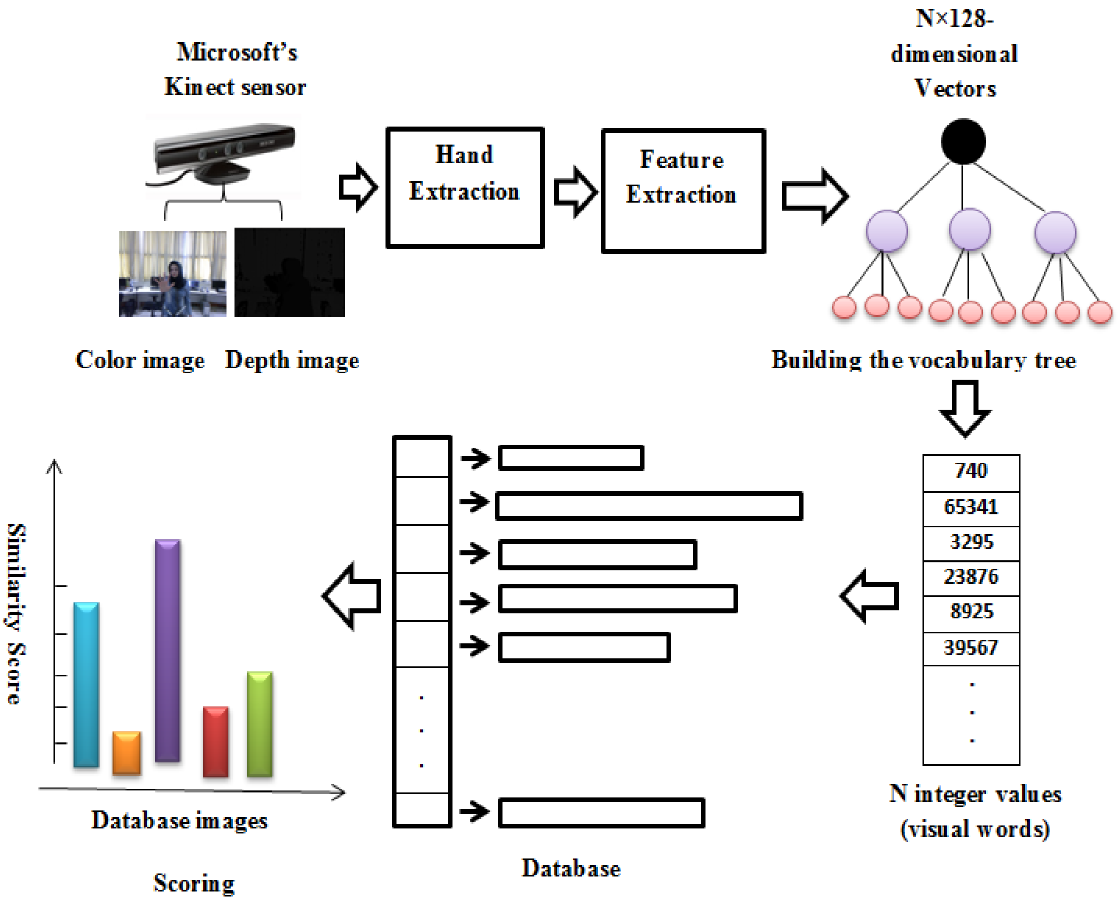

Figure 1. Illustrating overall picture of the system

2.2. Features Extraction using Scale Invariant Feature Transform

The main features of the SIFT algorithm which motivate us to apply this algorithm, are invariant to scale and rotation and real time extraction for low resolution images. The SIFT algorithm extract features in four stages:

First stage: A set of difference of Gaussian filters applied at different scales all over the image, and then the locations of potential interest points in the image are computed.

Second stage: The potential points are improved by removing points of low contrast.

Third stage: Assigning an orientation to each key point based on local image features.

Fourth stage: Computing a local feature descriptor at each key-point which is based on the local image gradient, transformed according to the orientation of the key-point to provide orientation invariance.

The extracted feature vectors from each hand image in this step are used to train our hand gesture recognition system. The number of key-points is dependent on the area of the detected hand. In fact, the 'five' gesture, the largest area gesture, has the maximum number of key-points. In this work, the number of the key-points which is obtained from the training images is from 30 to 78 key-points and each key-point has 128-dimensional feature vectors.

To have accurate digits (numbers) recognition system, we used several training images from different people with different scale, orientations, and illumination conditions. Therefore, the training stage is time consuming state. However, this will not affect the testing stage speed. In the next section, we discussed about the training of the vocabulary tree using the features vectors.

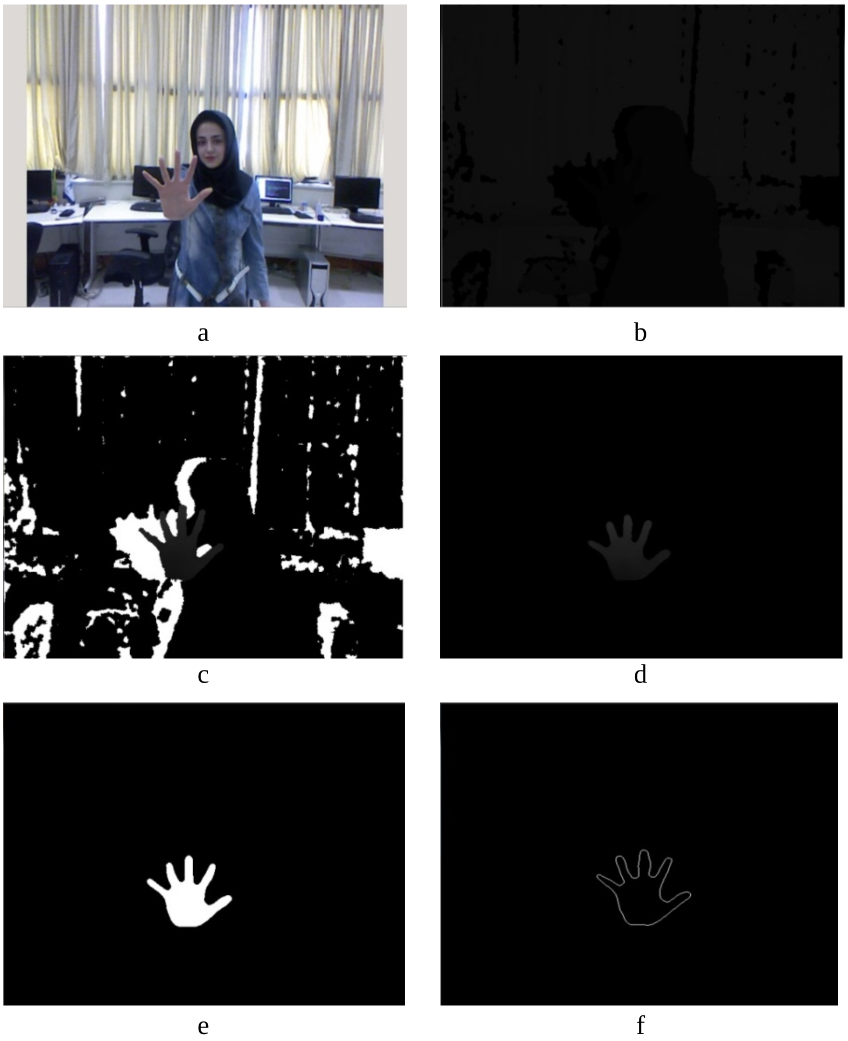

Figure 2. (a) Original color Image in clutter background (b) Depth Image.
(c) Rescaling Image (d) Median Filtering (e) Thresholding (f) Hand contour

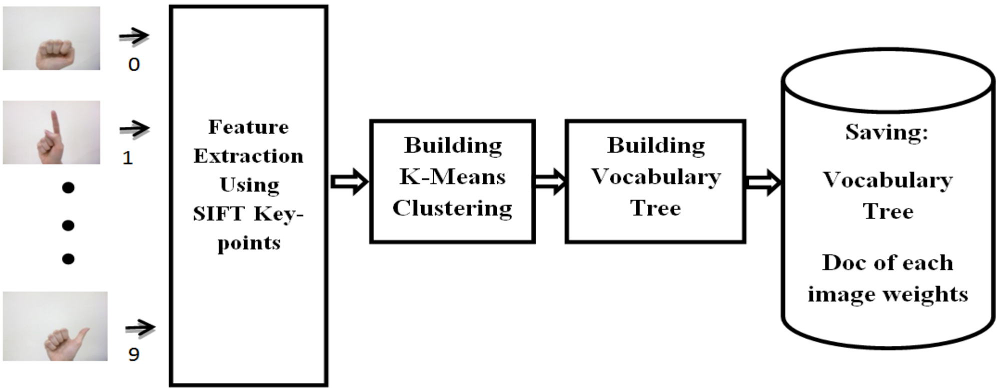
0
1
9
Feature Extraction Using SIFT Key-points
Building K-Means Clustering
Building Vocabulary Tree
Saving:
Vocabulary Tree
Doc of each image weights

Figure 3. Illustrating recognition procedure with vocabulary tree

3. OFFLINE TRAINING OF THE VOCABULARY TREE USING K-MEANS CLUSTERING

In this section, we implemented gesture classification for ten postures of finger using the vocabulary tree approach. The extracted SIFT features are hierarchically quantized in a vocabulary tree. In this way, each high-dimension feature vector is quantized into an integer which corresponds to a path in the vocabulary tree. We trained the vocabulary tree using 150 images from each hand gesture, which are “one”, “two”, “three”, “four”, “five”, “six”, “seven”, “eight”, “nine” and “ten” numbers. Our training images are captured from different people in various conditions to increase robustness of the classifier. Figure 4 depicts the training model. We first discuss about k-means clustering which is used to quantize feature space. Then, building a weighted vocabulary tree using TFIDF weighting and gesture recognition using k-nearest neighbor and voting are examined.

3.1. Building the Vocabulary Tree using k-Means Clustering

In the unsupervised training of the tree, firstly, the training data is clustered to the k = 4 centers using k-means method [19]. The training data is then divided into ten groups, where each group consists of the descriptor vectors closest to an individual cluster center. Afterwards, each group is segmented into in ten new parts by the k-means process. The process is continued to reach its maximum number of levels L = 6. Figure 4 shows the process of building the vocabulary tree. After building the vocabulary tree, we have to train it (assigning node weights) according to the data base. This is further detailed in the next section.

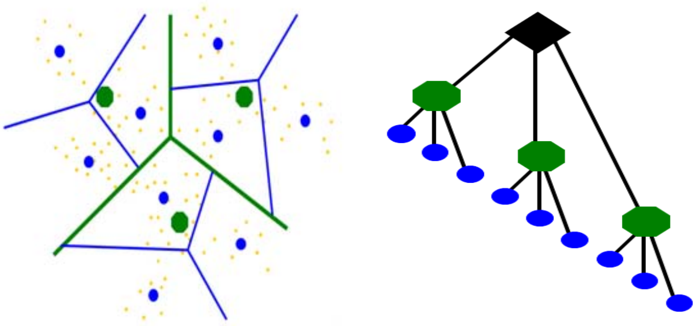

Figure 4. Illustrating process of building a vocabulary tree using k-means (k = 3) with branch factor 3 and three levels

3.2. Setting the TFIDF Weights of the Tree

Once the vocabulary tree is defined, we require to determine a weight wi to each node i in the vocabulary tree. Here, TFIDF, the product of term frequency and inverse document frequency, is used to assign the weights in the vocabulary tree. The tf-idf weighting algorithm diminishes weight nodes which appear often in the database as:

$$w_i = tf \times log\frac{N}{N_i} \tag{1}$$

Where N is the total number of images in the database, Ni is the number of images in the database with at least one key-point path through node i, and tf the frequency of occurrence of node i in place of Ni. We define query (qi = niwi) and database (di = miwi) vectors, where ni and mi are the number of key-points vectors of the query and database image, respectively, with a path through node i. After assigning the weights, scoring scheme is defined as:

$$Score(q,d) = \left|\frac{q}{|q|} - \frac{d}{|d|}\right| \tag{2}$$

Where | | is L1-norm.

In table 1, we investigate the effect of changing number of levels (L) and the branch factor (k) on the performance of the proposed algorithm. To this aim, we vary number of levels from 4 to 7, and the branch factor from 3 to 6. The results on the various modes indicated that the case of L = 6 and k = 4 is the best compromise between the accuracy and speed of training.

Table 1. Effect of Number of Levels and Branch Factor on the Algorithm Acuracy and Speed (Training time)

Branch Factor	*k = 3*		*k = 4*		*k = 5*		*k = 6*	
Num. of levels	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)
L = 4	5.29	71.25	5.14	88.24	6.58	78.78	7.96	75.98
L = 5	6.24	73.45	6.69	90.12	7.36	79.25	8.24	78.65
L = 6	6.96	75.25	**7.45**	**97.42**	8.45	80.59	9.54	79.33
L = 7	7.47	68.47	8.66	85.97	9.24	74.75	10.23	73.12

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed method, the proposed method is simulated on both our image data set and on public image data set. First, we present the experiment by simulation on our image data to show the performance of the proposed method in different situations. Then the method is simulated on public image data set to show its perceptual advantages in comparison with other methods.

We tested ten hand gestures as database image which are shown in Figure 5. Figure 5 simply shows detected hand gestures in the free background. The camera used for recording video files in our experiment is a Microsoft's Kinect which provides video capture with resolutions 640×480, at 30 frames-per second, which is adequate for real-time speed image recognition.

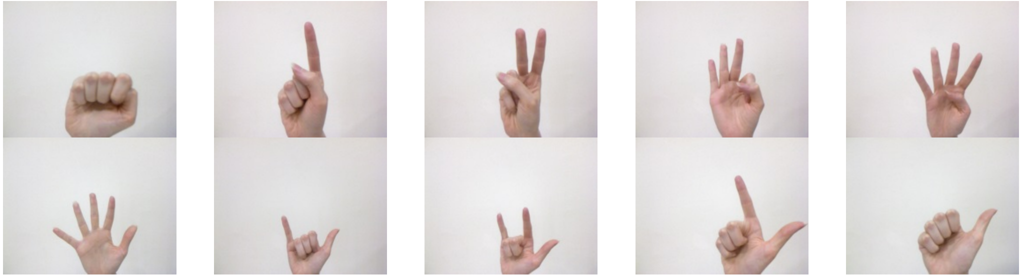

Figure 5. Ten detected hand gestures available in our image data set

The Sebastien Marcel database [20], which is a benchmark database in the field of hand gesture recognition, is also used as the public database image in this paper. This database contains (100 × 100 pixels) color images of six hand postures performed by different people against uniform and complex backgrounds. In the training stage, we captured 100 training images for each new hand posture in the clutter background.

The first experiment is performed on our database image to find recognition accuracy of the proposed method. Our method has excellent recognition results on these images as shown in Table 2. The overall recognition accuracy is 94.42 and the recognition time is about 15 milliseconds.

Figure 6. Our @Home robot in MRL at Qazvin Azad University

Table 2. Performance of the proposed method on our image database

Posture name	Recognition Accuracy	Recognition Time (second/frame)
One	98.25%	14.1 ms
Two	95.85%	14.3 ms
Tree	94.12%	16.2 ms
Four	95.23%	15.1 ms
Five	95.77%	15.3 ms
Six	94.42%	14.2 ms
Seven	96.12%	16.5 ms
Eight	95.33%	16.4 ms
Nine	97.85%	15.3 ms
Ten	96.23%	14.1 ms

As the second experiment, to examine the robustness of our method against different scales, rotations, and illuminations conditions, we test each gesture with several images which is captured in different situations. All the results are summarized in Table 3 for 1000 images. Recognition time is reported in millisecond.

Table 3. Performace of the proposed method for some typical gesture against scaling, rotation with diffrent illumination conditions

Gesture Name	Recognition Accuracy	Recognition Time (millisecond)
“two”	86.28%	86
“five”	83.21%	95
“seven”	81.33%	76
“eight”	75.56%	88
“ten”	76.14%	89

Finally, to compare our gesture recognition algorithm with other schemes, we used some papers [20-24], [2] and [12] that had a real-time performance. We report recognition time and accuracy (%) for The Sebastien Marcel database. As can be observed from the table, the proposed method outperforms other methods for recognition accuracy.

The proposed method is used in the @home robot which is an autonomous robot. This robot focuses on real-word application and can assist humans in everyday life. Figure 6 shows our robot in the mechatronics research laboratory (MRL) at Qazvin Azad University. Our experiments with @home robot verify that our algorithm is suitable for real-time applications. Some typical hand detected images in the clutter background with their feature points are shown in Figure 7. These posture are recognized in real-time by the vision section (Kinect Sensor in top of the head) of the robot and can be used as commands to control the robot.

Table 4. Comparison among our method and other real-time methods

Method	Number of posture	Background	Frame resolution	Recognition time	Recognition accuracy
[21]	15	Wall	160×120	0.4	94.89%
[22]	6	Cluttered	320×240	0.09-0.11	93.8%
[23]	3	Different	640×480	0.1333	96.2%
[12]	4	White wall	320×240	0.03	90.0%
[24]	8	Not discussed	640×480	0.066667	96.9%
[20]	6	Cluttered	100×100	Not discussed	76.1%
[2]	10	Cluttered	640×480	0.017	96.23%
our method	10	Cluttered	640×480	0.025	97.42%

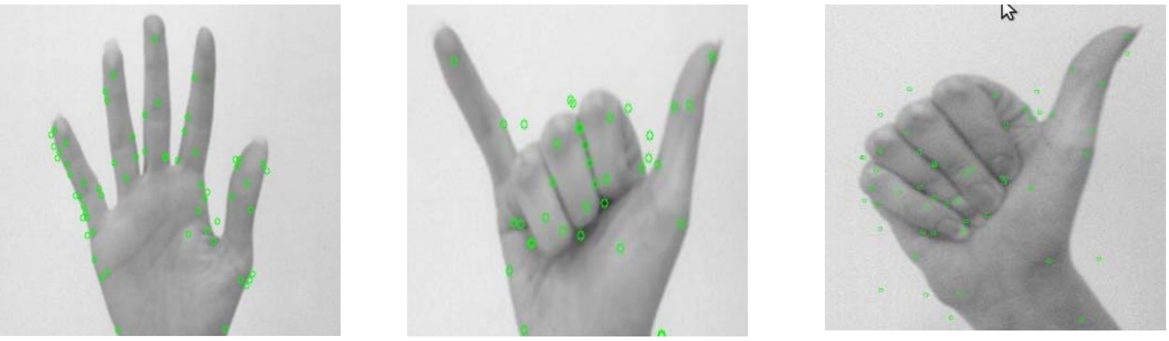

Figure 7. Some typical hand detected images in the clutter background with their feature points in small green circles

5. CONCLUSION

We proposed a method for hand gesture recognition using Microsoft's Kinect sensor. The Microsoft's Kinect which is an infrared camera captures the depth information and provides a gray scale depth image. According to the depth image, hand can be detected based on its distinct gray level. We found that hand detection process is independent from background and illumination when hand located nearer to the sensor from other objects. After hand detection, we trained a vocabulary tree to recognize hand gestures. We experimentally obtained the best parameters (number of levels and branch factor) of the vocabulary tree to have proper recognition accuracy. Extensive simulations on both our image dataset and the Sebastien Marcel database show the superiority of the scheme in comparison to other state-of-the-art. Finally, we implemented our algorithm on @home robot. The results confirm that the proposed method has a significant accuracy and can be used in real-time applications.

REFERENCES

[1] A El-Sawah, N Georganas, and E Petriu. “A prototype for 3-D hand tracking and posture estimation”. *IEEE Trans. Instrum. Meas*. 2008; 57(8): 1627–1636.

[2] Dardas NH, Georganas, Nicolas D. “Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques”. *IEEE Transactions on Instrumentation and Measurement*. 2011; 60(11): 3592, 3607.

[3] H Zhou and T Huang. “T*racking articulated hand motion with Eigen dynamics analysis*”. in Proc. Int. Conf. Comput. Vis. 2003; 2: 1102-1109.

[4] JM Rehg and T Kanade. “*Visual tracking of high DOF articulated structures: An application to human hand tracking*”. in Proc. Eur. Conf. Comput. Vis. 1994: 35-46.

[5] AJ Heap and DC Hogg. “T*owards 3-D hand tracking using a deformable model*”. in Proc. 2nd Int. Face Gesture Recog. Conf., Killington, VT. 1996: 140-145.

[6] Y Wu, JY Lin, and TS Huang. “*Capturing natural hand articulation*”. in Proc. 8th Int. Conf. Comput. Vis., Vancouver, BC, Canada. 2001; II: 426-432.

[7] B Stenger, PRS Mendonça, and R Cipolla. “*Model-based 3D tracking of an articulated hand*”. in Proc. Brit. Mach. Vis. Conf., Manchester, U.K. 2001; I: 63-72.

[8] B Stenger. “*Template based hand pose recognition using multiple cues*”. in Proc. 7th ACCV. 2006: 551-560.

[9] L Bretzner , I Laptev and T Lindeberg. “*Hand gesture recognition using multiscale color features, hieracrchichal models and particle filtering*”. Proc. Int. Conf. Autom. Face Gesture Recog. 2002

[10] A Argyros and M Lourakis. “*Vision-based interpretation of hand gestures for remote control of a computer mouse*”. Proc. Workshop Comput. Human Interact. 2006: 40 -51.

[11] C Wang and K Wang. Hand Gesture Recognition Using Adaboost With SIFT for Human Robot Interaction. Springer-Verlag. 2008; 370.

[12] A Barczak and F Dadgostar. “Real-time hand tracking using a set of co-operative classifiers based on Haar-like features”. Res. Lett. Inf. Math. Sci. 2005; 7: 29-42.

[13] Q Chen, N Georganas and E Petriu. “*Real-time vision-based hand gesture recognition using Haar-like features*”. Proc. IEEE IMTC. 2007: 1-6.

[14] DG Lowe. “Distinctive image features from scale-invariant keypoints”. *Int. J. Comput. Vis*. 2004; 60(2): 91-110.

[15] L Juan and O Gwun. “A comparison of SIFT, PCA-SIFT and SURF”. *Int. J. Image Process. (IJIP)*. 2009; 3(4): 143-152.

[16] Y Ke and R Sukthankar. “*PCA-SIFT: A more distinctive representation for local image descriptors*”. in Proc. IEEE Conf. Comput. Vis. Pattern Recog. 2004: II-506–II-513.

[17] H Bay, A Ess, T Tuytelaars, and L Gool. “SURF: Speeded up robust features”. *Comput. Vis. Image Understand. (CVIU)*. 2008; 110(3): 346-359.

[18] Microsoft Corp. Redmond WA. Kinect for Xbox 360.

[19] DJC MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[20] S Marcel. “*Hand posture recognition in a body-face centered space*”. in Proc. Conf. Human Factors Comput. Syst. (CHI). 1999: 302-303.

[21] W Chung, X Wu, and Y Xu. “*A real time hand gesture recognition based on Haar wavelet representation*”. in Proc. IEEE Int. Conf. Robot.Biomimetics. 2009: 336-341.

[22] Y Fang, K Wang, J Cheng, and H Lu. “*A real-time hand gesture recognition method*”. in Proc. IEEE Int. Conf. Multimedia Expo. 2007: 995-998.

[23] L Yun and Z Peng. “*An automatic hand gesture recognition system based on Viola-Jones method and SVMs*”. in Proc. 2nd Int. Workshop Comput. Sci. Eng. 2009: 72-76.

[24] Y Ren and C Gu. “*Real-time hand gesture recognition based on vision*”. in Proc. Edutainment. 2010: 468-475.

BIOGRAPHIES OF AUTHORS

Mino Hamissi received the B.Sc. degree from Qazvin Azad University, Qazvin, Iran, in 2009, where sheis currently pursuing the M.Sc. degree, all in computer engineering.

Karim Faez was born in Semnan, Iran. He received his BSc. degree in Electrical Engineering from Tehran Polytechnic University as the first rank in June 1973, and his MSc. and Ph.D. degrees in Computer Science from University of California at Los Angeles (UCLA) in 1977 and 1980 respectively.

Professor Faez was with Iran Telecommunication Research Center (1981-1983) before Joining Amirkabir University of Technology (Tehran Polytechnic) in Iran in March 1983, where he holds the rank of Professor in the Electrical Engineering Department.

He was the founder of the Computer Engineering Department of Amirkabir University in 1989 and he has served as the first chairman during April 1989-Sept. 1992. Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, Research and Technology (during 1988-1996). His research interests are in Biometrics Recognition and authentication, Pattern Recognition, Image Processing, Neural Networks, Signal Processing, Farsi Handwritten Processing, Earthquake Signal Processing, Fault Tolerance System Design, Computer Networks, and Hardware Design.

Dr. Faez coauthored a book in Logic Circuits published by Amirkabir University Press. He also coauthored a chapter in the book: Recent Advances in Simulated Evolution and Learning, Advances in Natural Computation, Vol. 2, Aug.2004,World Scientific. He published about 300 articles in the above area. He is a member of IEEE, IEICE, and ACM, a member of Editorial Committee of Journal of Iranian Association of Electrical and Electronics Engineers, and International Journal of Communication Engineering. Emails: kfaez@aut.ac.ir, kfaez@ieee.org, kfaez@m.ieice.org