Hand Gesture Recognition Using Kinect's Geometric and HOG Features

Hamda Marouane
LIMIARF, Faculty of Science,
Mohammed V University
Rabat
Morocco
hamda.marouane@gmail.com

Abdelhak Mahmoudi
LIMIARF, Ecole Normale Supérieure,
Mohammed V University
Rabat
Morocco
abdelhak.mahmoudi@gmail.com

ABSTRACT

Hand gesture recognition plays an important role in human computer interaction (HCI). Despite the recent progress, the accuracy of up-to-date methods is still not satisfactory. In this work, we proposed a comparative study to recognize six hand gestures in real time using the Kinect sensor. First, we developed a tracking method of the hand in the scene where the center of the palm is detected using depth data and projected into the color image. Second, geometric features were extracted from depth image and Histogram of Oriented Gradients (HOG) descriptors from the color image. Finally, based on those extracted features, a support vector machines (SVM) and an artificial neural network (ANN) are trained and compared.

CCS CONCEPTS

• **Human-centered computing** → Gestural input;

KEYWORDS

HCI, Hand Gesture Recognition, HOG, SVM, ANN

ACM Reference format:

M. Hamda, A. Mahmoudi, 2. In *Proceedings of BDCA conference,* Tetuan, Morocco*, Mars 2017 (*BDCA'17*)*, 4 pages.
DOI:

1 INTRODUCTION

Human computer interaction (HCI) received a great deal of attention this last decade. The interest of this topic arises from the need to develop more intuitive mechanisms to interact with computers.

Several systems have been developed for the human-machine interface such as keyboard, mouse, joystick, speech, etc. But these types of interfaces have shown their limits in degrees of freedom. Last researches has turned to more natural interactions systems. This is the case for gestural human-machine interfaces. Electronic gloves are among the most expressive interfaces to detect gestures. These sensors provide the position of the hand and fingers joints angles. However, they are costly and cumbersome.

Fortunately, with the technical advances and the appearance of inexpensive cameras, it is now possible to develop real time gesture recognition systems based on computer vision. Furthermore, thanks to the recent development of inexpensive depth cameras such as the Kinect sensor, new opportunities for hand gesture recognition (HGR) emerge. Instead of wearing a data glove, the use of the Kinect sensor can also detect and segment the hands robustly.

In spite of many recent successes in applying the Kinect sensor to articulated face recognition [1] and human action recognition [2], it is still an open problem to use Kinect for HGR. Many vision-based approaches have been proposed in the literature. In [3], Ren et al. introduces an HGR system using the Kinect sensor based on both depth and color information for hand detection. The system uses the new metric shape distance called Finger Earth Mover's Distance (FEMD) typically designed for hand shapes. Lang et al. [4], proposes a system of general HGR using Kinect based on Hidden Markov Model (HMM). Rekha et al. [5] proposes a hybrid approach that combines the local descriptors (SURF and SIFT) and the Hu Invariant Moment global descriptor. Hiyadi et al. [6] proposed a 3D dynamic HGR method for human robot interaction based on depth information provided by the Kinect. The body is tracked using the skeleton algorithm provided by the Kinect SDK. The main idea of theire work is to compute the angles of the upper body articulations which are active when executing gesture. The variation of these angles is then used as inputs for an HMM in order to recognize the dynamic gestures. In [7], Ramirez et al. developed an HGR method based on the features extracted from the geometric properties of the hand.

In this paper, we propose a comparative study to recognize six hand gestures in real time using the Kinect sensor. Initially, based on the image processing techniques, we developed a method of hand tracking. Next, we proposed two approaches to extract features from the detected hand. The first have a structure similar to the one proposed in [7] with an improvement in the number of features. The second approach is based on HOG descriptor (Histogram of Oriented Gradient). Finally, we have classified the features obtained from the two methods by two machine learning algorithms, support vector machines (SVM) and artificial neural networks (ANN).

2 HAND GESTURES RECOGNITION

Our HGR system consists of the following steps: (1) data acquisition from Kinect sensor, (2) hand detection and tracking, (3) features extraction, (4) gestures classification and display the result as text (Fig 1).

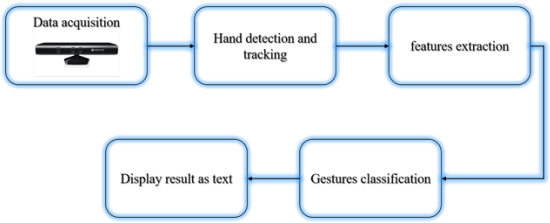

Figure 1: Diagram of the proposed hand gestures recognition process

2.1 Hand Detection and Tracking

The subject is assumed to be in front of the Kinect sensor providing the depth image. This last is then segmented assuming the gesture is performed at a minimum distance of about 0.8m and maximum of 1m (Fig. 2).

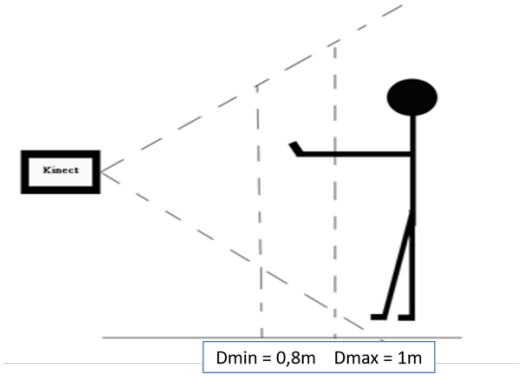

Figure 2: Depth interval detection.

After eliminating the small regions by morphological operators, the contour ζ of the hand is detected allowing the extraction of the convex hull and the bounding rectangle of the hand (Fig. 3).

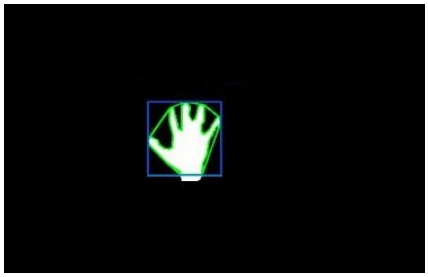

Figure 3: Hand tracking in the depth image. In green: the convex hull. In blue, the bounding box.

The center of the palm is one of the keys providing the position of the hand. The center C of the hand can be considered approximately as the center of the largest circle corresponding to the inside of the hand contour ζ (Fig. 4).

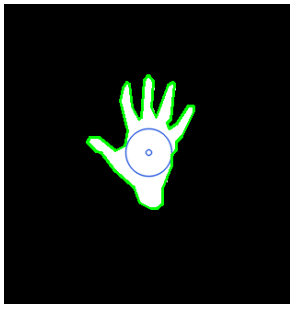

Figure 4: Detection of the hand's center inside the contour. The blue circle is the largest circle inside the detected contour.

In order to find this center, we compute the maximum of the minimum distances of each point $P_I\ (i)$ inside each point $P_\zeta\ (j)$ of the contour using the following equation:

$$\mathrm{D}\left(P_I(\mathrm{i}), P_\zeta(\mathrm{j})\right) = \sqrt{\left(P_I^x(i) - P_\zeta^x(j)\right)^2 + \left(P_I^y(i) - P_\zeta^y(j)\right)^2} \quad (1)$$

Where i = 1… N_I and j = 1...N_ζ and N_I , N_ζ are the number of points inside the hand and those belonging to the contour ζ respectively.

To be able to track the hand in the color image, the detected center has to be projected on it. However, this projection can only be performed after calibrating the Kinect. Our calibration method is based on the detection of the centers of disks in the RGB image and the depth image of the Kinect (Fig.5).

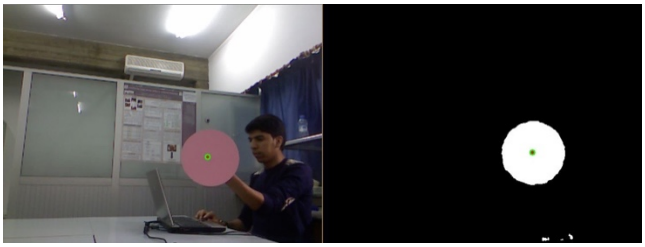

Figure 5: Calibration of the Kinect. The offset is the difference between positions of the center of the disk detected in both depth and color images.

After recording n disk centers in both images, $C_{rgb}(i)$ for the color image and $C_{depth}(i)$ for the depth image, the offset d can be defined as the mean of the absolute difference of the two centers as shown by the following equation:

$$\mathrm{d} = \frac{1}{n}\sum_{i=0}^{n} |C_{rgb}(i) - C_{depth}(i)| \quad (2)$$

After the calibration phase, we projected the center of the hand C and the delimiting rectangle on the color image with respect to the computed offset d.

2.2 Features Extraction

Geometric features. In this approach we have modeled the gestures using features extracted from the geometric properties of the hand. We used a method similar to that presented in [7] with an improvement in the number of features. A graphical description of the geometric properties used to calculate the features characterizing the static gestures is shown in Fig. 6. The used vocabulary is shown in Fig. 7.

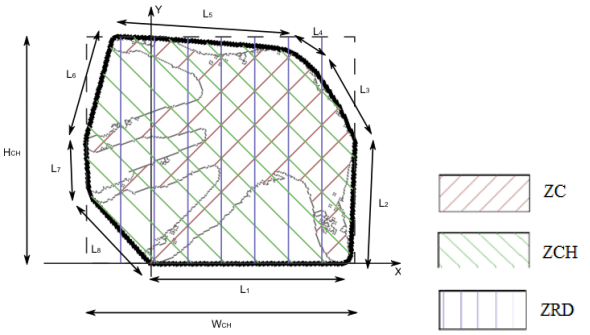

Figure 6: Hand's geometric features as reported from [7].

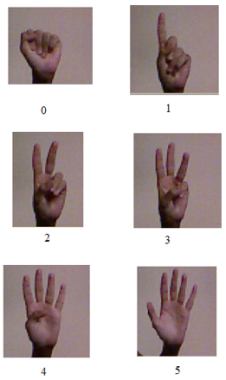

Figure 7: Gestures vocabulary with class labels

Therefore, the used geometric features vector is as follows:

V = [ZC/SCH; SCH/ZCH; L_{max}/SCH; WCH/HCH; ZCH/ZRD; ZRD/WCH; ZRD/HCH; TC/SCH]

Where:

- ZC: Contour area
- ZCH: Convex hull area
- ZRD: Bouding rectangle area
- SCH = $\sum L_i$
- HCH: Bounding rectangle height.
- WCH: Bounding rectangle width.
- L_{max} = max(L_i)
- TC: Size of contour

We recorded 500 examples for each gesture. 400 for training and 100 for test. In total we obtained a vector of (500x6) x8 examples.

Histogram of Oriented Gradient (HOG). HOG is a descriptor used in computer vision for object detection [8]. The technique counts occurrences of gradient orientation in localized portions of an image. It is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.
In order to extract HOG features, the color image corresponding to the tracked hand is fist resized to fit 64x128 pixels, then divided into 16x16 blocks with a 50% overlap. This yields to 105 (7x16) blocks in total. Each block should consist of 2x2 cells with a size of 8x8 as shown in Fig. 8. Finally, the obtained gradient orientations vector is quantified using 9 bins. We then end with a vector of 3780 features (105x4x9).

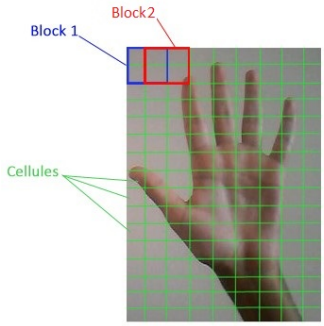

Figure 8: Example of dividing an image into blocks and cells using HOG algorithm.

2.3 Classifiers

Support Vector Machines (SVM). SVM is a supervised machine learning method aiming mainly to simultaneously maximize the generalization performance and the geometric margin between the classes [9]. It was observed that the use of kernel functions solves the problem of high dimensionality allowing SVMs to handle nonlinearity in an efficient way.
In this paper we used SVM classifier with two kernels: The radial basis function (RBF) kernel and Chi² kernel defined respectively in the two following equations:

$$k(x, x') = exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (3)$$

$$k(x, x') = exp\left(1 - \sum_{i=1}^{n} \frac{2(x - x')^2}{x + x'}\right) \quad (4)$$

Artificial Neural Network (ANN). ANN is a connected architecture of single layered perceptrons allowing to classify non linearly sets of data. The most commonly used architecture is the Feed-forward network in which the signals are allowed to travel

one way only from an input layer to an output layer through one or more hidden layers. The input layer receives the features vector x and the output layer contains its corresponding class y. Every unit in the input layer sends its activation value (Eq. 5) to each of its connected units in the hidden layer. Each of these hidden units calculates its own activation value and propagate it to the next hidden layer. The last resulted signal is then passed on to the units in the output layer.

$$y = g\left(\sum_{j=1}^{n} w_j x_j - b\right) \quad (5)$$

Where w is the network parameters, g is the sigmoid function and b is the bias.

Receiver Operating Characteristic (ROC) curve. The ROC curve is formed using the confusion matrix (Fig. 9) by plotting true positive rate (TPR) over false positive rate (FPR) defined as follows:

$$\text{TPR} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (7)$$

		True Class: p	True Class: n
Predicted Class	p	TP (True Positive)	FP (False Positive
	n	FN (False Negative)	TN (True Negative)

Figure 9: Confusion matrix

In order to assess different classifier's generalization performances, one generally uses the area under the ROC curve (AUC) as an evaluation metric. Smith et al. [10] have shown that the AUC is a better measure of classifier generalization performance than the accuracy measure. Indeed, AUC has an important statistical property: it is equivalent to the probability that the classifier will evaluate a randomly chosen positive example higher than a randomly chosen negative example. AUC metric is applied just in the case of binary classification. We have then converted our multi-class classification problem to a binary classification using the principle *one class versus all*.

Cross validation. To classify the obtained features of the two methods (Geometric and HOG) we used the two algorithms of machine learning SVM and ANN and computed the AUC of each of them. To ensure a generalized model, we have applied the cross-validation method using 8-folds and chose the best parameters for each classifier. The AUC values are obtained using the best parameters for SVM (hyper-parameter C and σ value of the RBF kernel) and the best number of units in each hidden layer for the ANN classifier.

3 RESULTS AND DISCUSSION

3.1 Results using geometric features

In this method we used ANN with 3, 4 and 5 layers and SVM with the two types of kernels RBF and CHI². The results of this method are shown in Fig. 10. We can clearly see that the use of the ANN classifier gives always the best results regardless of the number of layers. One can then choose to use three layers for fast recognition. The Chi² kernel gives also good results except for the class 3. This can be explained by the fact that this class can be easily confused with the class 2 and 4. The RBF kernel is the worst perhaps because of the parameter σ tuning.

Figure 10: AUC of the classification of the geometric features using SVM with two kernels (RBF and CHI²) and ANN with 3, 4 and 5 layers.

3.2 Results for HOG features

In this method we did not use the ANN algorithm because it is too slow in the learning phase. So we just use the linear SVM. The results of this method are shown in Fig. 11.

Figure 11: AUC of the classification of HOG features using linear SVM.

4 CONCLUSION

In this paper, we proposed a comparative study to recognize six hand gestures in real time using the Kinect sensor. Geometric and Histogram of Oriented Gradients features are extracted from the depth and color images and classified using Support Vector Machines with two kernels (Radial Basis Function and Chi²) and an Artificial Neural Network with 3, 4 and 5 layers. The generalization performance is computed using the Area under the Receiver Operating Characteristics curve with 8-fold cross validation. We concluded that the ANN algorithm is very efficient with geometric features, but too slow when we have a large vector of features as it is the case of HOG features.

REFERENCES

[1] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, "3D deformable face tracking with a commodity depth camera," in Proc. Eur. Conf. Computer Vision, Crete, Greece, 2010.

[2] Y. Choubik and A. Mahmoudi, "Machine Learning for Real Time Poses Classification Using Kinect Skeleton Data Sign in or Purchase", in Proc. IEEE Conf. Computer Graphics, Imaging and Visualization, Morocco, 2016.

[3] Z. Ren, J. Meng, and J. Yuan. "Depth Camera Based Hand Gesture Recognition and its Applications in Human-Computer-Interaction". In Proc. IEEE (ICICS), 2011.

[4] S. Lang, "Sign Language Recognition with Kinect", Thesis, Berlin university, 2011.

[5] J. Rekha, "Hand Gestures Recognition for Sign Language: A new hybrid approach", In Proc. IPCV Conf. Computer Vision and Pattern Recognition, 2011.

[6] H. Hiyadi. F.E. Ababssa, E.H. Bouyakhf, C. Montagne and F. Regragui, "Reconnaissance 3D des Gestes pour l'Interaction Naturelle Homme Robot", in Proc. 15m edition des journes francophones des jeunes chercheurs en vision par ordinateur (ORASIS), Jun 2015.

[7] V. Ayala-Ramirez "A Hand Gesture Recognition System Based on Geometric Features and Color information for Human Computer Interaction Tasks", In Proc. ROSSUM Conf. Computer Vision, 2011

[8] N. Dalal, N. and B. Triggs, "Histograms of Oriented Gradients for Human Detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, San Diego, CA, USA.

[9] V. N Vapnik. The nature of statistical learning theory, volume 8. Springer, 1995.

[10] S. M. Smith and T. E. Nichols. Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference. NeuroImage, vol. 44, no. 1, pages 83–98, 2009.