2014 International Conference on Advanced Technologies for Communications

(ATC 2014)

Hanoi, Vietnam
15-17 October 2014

IEEE

IEEE Catalog Number:	CFP14ATC-POD
ISBN:	978-1-4799-6957-9

Table of Contents

Communications

Electronics

Poster 2

Signal Processing

Dynamic hand gesture recognition using RGB-D motion history and kernel descriptor

Thanh-Hai Tran, Ta-Hoang Vo, Duc-Tuan Tran, Thi-Lan Le
International Research Institute MICA, HUST
CNRS/UMI 2954 - Grenoble INP
Hanoi University of Science and Technology

Thuy Thi Nguyen
Faculty of Information Technology
Vietnam National University of Agriculture

Abstract

Gesture recognition has important applications in sign language and human - machine interfaces. In recent years, recognizing dynamic hand gesture using multi-modal data has become an emerging research topic. The problem is challenging due to the complex movements of hands and the limitations of data acquisition. In this work, we present a new approach for recognizing hand gesture using motion history images (MHI) [1] and a kernel descriptor (KDES) [2]. We propose to use an improved version of MHI for modeling movements of hand gesture, where MHI is computed on both RGB and depth data. We propose some improvements in patch-level feature extraction for KDES, which is then applied to MHI to represent gesture features. Then SVM classifier is trained for recognizing gestures. Experiments have been conducted on challenging hand gesture data set of CHALEARN contest [3]. An extensive investigation has been done to analyze the performance of both improved MHI and KDES on multi-modal data. Experimental results show the state-of-the-art of our approach in comparison to the results of the contest.

Keywords — dynamic gesture recognition, motion analysis, kernel descriptor

I. INTRODUCTION

Gesture is an intuitive and efficient mean of communication between human and human in order to express information or to interact with environment. In Human Computer Interaction (HCI), hand gesture can be an ideal way that a human controls or interacts with a machine. In that case, machine must be able to recognize human hand gestures. Recently, hand gesture recognition becomes a hot research topic in the HCI and Computer Vision field due to its wide applications, such as sign hand language, computer game, e-learning, human-robot interaction.

Vision-based approaches for hand gesture recognition use one or several cameras to capture sequence of images of hand gestures. The problem is challenging because of the following reasons. Firstly, as hand posture has at least 27 Degrees of Freedom (DoF), the number of hand postures to be recognized is numerous that requires a lot of examples for training a classification model. Secondly, the location of camera should be chosen so that it can observe the entire hand gestures. This one is difficult because hand can occlude itself. Finally, recognizing correctly hand gesture in images is time consuming that makes it hard to develop real-time applications.

Recently, Microsoft has launched Kinect sensor and it soon become a common device in many areas including computer vision, robotics human interaction and augmented reality. The most advantage of this device is its low-cost while providing depth information of the scene. Depth data is invariant to lighting changes. This property attracts lots of researchers working with depth data as a complement of RGB data.

A well-known event organized recently that drawn a lot of attentions in the field is the CHALEARN contest [1]. This is a contest on gesture and sign language recognition from video data organized by Microsoft. Last year, the contest focused on hand gesture recognition using multimodal information coming from RGB-D and also audio sensors. There are 54 participants in the CHALEARN with 17 submissions. The used modalities are various combinations of audio, RGB, Depth and Skeleton. Most of the participants used ordinary techniques to extract features from the multimodal data, and traditional machine learning techniques were employed for training a classifier for recognition. It turned out that audio data contributes significantly to recognizing gestures. However, using audio data is not typical in gesture recognition and in many situations it may not be available.

In this paper, we present a new approach on hand gesture recognition using visual cues and depth information. We investigate how recent proposed techniques for feature extraction can be used for this kind of multimodal data. Due to the characteristic of dynamic hand gesture, we propose to model the motion information using motion history image (MHI). We then represent each video shot (one dynamic hand gesture) by a MHI. The kernel descriptor KDES has shown to be the best descriptor until now for image classification [2]. We propose some improvements in patch-level feature extraction for KDES, which is then applied to MHI to compute features for gesture representation. Support Vector Machine (SVM) is used for hand gesture classification. Moreover, we will analyze deeply the characteristic of MHI as well as Kernel descriptor on a benchmark dataset CHALEARN with different information channels (RGB, Depth and combining of both).

The remaining of this paper is organized as follows. In section II we present related works on hand gesture recognition using multimodal information from Kinect sensor in general Depth and RGB- in particular. Section III explains the general framework and details each step of our proposed method. Section IV describes experimental results. Section V concludes and gives some ideas for future works.

II. Related Works

Many methods have been proposed for hand gesture recognition. A survey of the methods can be seen in [4]. In this section, we will review some works in the context of CHALEARN contest because they are closely related to our work. In the following we will briefly present some methods that have been published recently in the ICMI workshop [1]. The reasons are: i) In these works new techniques have been proposed, evaluated and compared to the state-of-the-art techniques, so they are up-to-date methods; ii) these methods have been evaluated on the CHALEARN database, which we will use to test our approach.

With 54 participants and 17 submissions, proposed approaches employed various combinations of modalities, including audio, RGB, Depth and Skeleton. Participants used mostly classical features extraction techniques. Traditional machine learning techniques were employed for training a classifier, including Hidden Markov Model (HMM), K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF).

Hu et al [5] proposed to fuse features extracted from different data types including audio and skeletal information. Mels Frequency Cepstral Coefficients (MFCCs) are audio features that will be used in HMM for classification. They also use skeletal data, extract 3D coordinates of 4 joints to make up a feature vector of 12 dimensions. KNN is used to decide which category the hand gesture belong to. The similarity between two hand gestures is computed with using Dynamic Time Warping (DTW). Late fusion is used to combine the recognition results from the two classifiers. This method is ranked the first in the contest with a test score is of 0.127.

Bayer et al. [6] also used audio and skeletal data for hand gesture representation. From skeletal data, each skeletal joint contains 9 coordinates: world position, pixel position and world rotation. Only 14 per 20 joints above waist are considered that make 126 time series per gesture. They then use 4 summary statistics to aggregate each of 126 values that gives 504 dimensions feature vector for one gesture. Extremely Randomized Tree is used to learn and to recognize hand gesture based on skeletal representation. Concerning audio data, 13 first MFCCs are used to characterize speech signal. Then, two classifiers, Gradient Boosting Classifier and RF, are trained on this descriptor. Finally, weighted technique is used for model averaging. This method is ranked the third in the contest with a test score is of 0.168.

Chen et al. [7] proposed a method for hand gesture recognition using skeletal and RGB data. Two kinds of features are extracted from the skeletal data that are normalized 3D joint position and the pair wise distances between joints. In addition, Histogram of Oriented Gradients (HOG) features are extracted on the left and right hand regions. These features are concatenated to form a description of the hand gesture. Finally, extreme learning machine (ELM) technique was used for classification.

Nadakumar et al. [8] proposed a method to combine different information (audio, video, skeletal joint) for hand gesture representation. For the audio information, 36 MFCCs are used with HMM to classify hand gesture. 3D coordinates of 20 skeletal joints are used to make 60 dimensional frame vector. A covariance matrix will be computed from all frames of the video shot. All elements (1830) above the main diagonal of the matrix are considered as descriptor for hand gesture. Typical Support Vector Machine (SVM) is used to identify gesture based on covariance vector. For RGB video, they extract STIP (Space Time Interest Point) descriptor. Bag of Word (BoW) and SVM are used to represent and recognize hand gestures. This method is ranked the seventh in the contest with test score is of 0.244.

One can see that, as we mentioned above, the participants of the CHALEARN mostly used traditional techniques for features extraction and traditional machine learning techniques for learning the classifier. None of them has explored the simple yet efficient MHI for motion representation and attempted to combine it with the state-of-the-art kernel descriptor KDES. These will be investigated in our work.

III. Proposed Approach

A. General description

We propose a framework for hand gesture recognition that consists of two phases: learning and recognition. We could see the main steps of the framework in the Fig. 1. In general, as well as in CHALEARN contest, RGB-D data could be acquired using a Kinect sensor.

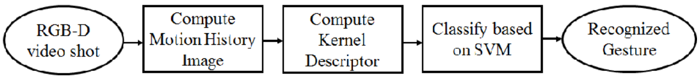


Fig.1: Main steps of the proposed method for dynamic hand gesture recognition

1. **Compute MHI:** As a dynamic hand gesture is a sequence of consecutive frames, we propose to represent each video shot containing one dynamic hand gesture by a MHI computed from this frame set.
2. **Feature extraction:** Kernel descriptor has been shown to be the best features for object and image classification [2]. We would like to evaluate this features on MHI image. As the best of our knowledge, there are no work on the combination of kernel descriptor with MHI for dynamic hand gesture recognition.
3. **Model learning:** In function of extracted features, a compatible recognition model will be chosen. We propose to use Support Vector Machine (SVM).
4. **Recognition:** Finally to evaluate the methods, we test all examples in the testing data using learnt models previously.

In the following, we will present in detail each step in the overall system.

B. Computation of Motion History Image

Motion history image is a simple but efficient technique to describe movements. It has been widely used in action recognition, motion analysis and other related applications. Due to these reasons, we extract MHI to serve as action descriptor. In addition, in [9], the authors have shown that using backward and forward MHI could improve significantly the performance of recognition. Forward MHI (fMHI) encodes forward motion history while backward MHI (bMHI) encodes the backward motion history. Therefore, we will consider MHI, backward and forward MHI for gesture representation.

1) Motion History Image

In an MHI, pixel intensity is a function of the motion history at that location, where brighter values correspond to more recent motion. This single image contains the discriminative information for determining how a person has moved (spatially and temporally) during the action. Denoting I(x, y, t) as an image sequence, each pixel intensity value in an MHI is a function H_τ^I of the temporal history of motion at that point, namely:

$$H_\tau(x,y,t)=\begin{cases}\tau & if\ \psi(x,y,t)\neq 0\\ 0 & if\ \psi(x,y,t)=0 \quad and \quad H_\tau(x,y,t-1)<\tau-\delta\\ H_\tau(x,y,t-1) & otherwise\end{cases} \tag{1}$$

Here, *(x,y)* and *t* show the pixel position and time, Ψ (x,y,t) is the object's presence in the current video image, the duration τ decides the temporal extent of the movement (in terms of frames), and δ is the decay parameter. The remaining timestamps in the MHI are removed if they are older than the decay value τ- δ. This update function is called for every new video frame analyzed in the sequence. The result of this computation is a scalar-valued image where more recently moving pixels are brighter and vice-versa. Here is the description of the Ψ function:

$$\psi(x,y,t)=\begin{cases}1 & if\ D(x,y,t)\geq\xi\\ 0 & otherwise\end{cases} \tag{2}$$

where D(x,y,t) is defined with the difference distance Δ:

$$D(x,y,t) = |I(x,y,t) - I(x,y,t \pm \Delta)|$$

Actually, we calculate the difference between two consecutive frames. At each pixel, if value of it is large enough, then there is a motion; by contrast, there is no motion. Here the brightness of a pixel corresponds to its recency in time (i.e. brightness of a pixel are the most current timestamps) (Fig. 2). Parameter δ has effect to result of MHI. Depending on the value chosen for the decay parameterδ, an MHI can encode a wide history of movement (Fig. 2).

One problem that we need take into account is that for a video shot, the starting and the stopping time of the gesture could be very different from person to person. When the person stops soon and returns to resting state, if we take all the sequence to compute MHI, then the MHI could forget all previous motions and contains only motionless information. Therefore, before computing MHI or bMHI and fMHI, we look for the resting position and MHI, bMHI, fMHI will be computed only until this resting position.

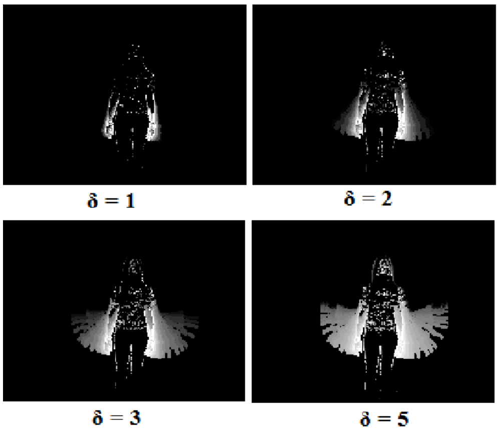


Fig.2: Effect of altering the decay parameter δ (in seconds)

To do this, we compare the difference in energy of the current frame with the end frame then define the resting at the position when the energy is lower than 2/3 the maximal value and does not change significantly any more. Fig. 3 illustrates the difference in energy.

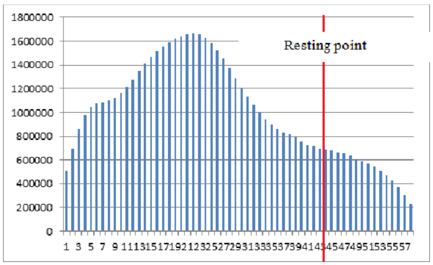


Fig.3: Difference in energy (sum all pixel values in image) between each frame and the end frame of the sequence. Horizontal axis represents consecutive frame in the sequence. Vertical axis represents the difference in energy.

2) Backward MHI

Backward MHI is defined similar to MHI:

$$H_\tau^b(x,y,t)=\begin{cases}\tau & if\ \psi(x,y,t)=1\\ Max(0,H_\tau^b(x,y,t)-\delta) & otherwise\end{cases} \tag{3}$$

but the threshold function is replaced by:

$$\psi(x,y,t)=\begin{cases}1 & if\ D(x,y,t)\leq-\xi\\ 0 & otherwise\end{cases} \tag{4}$$

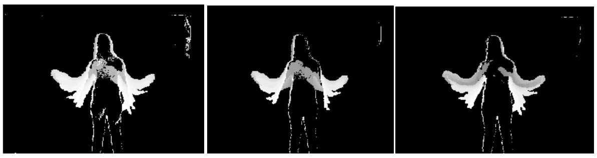
with D(x,y,t) = I(x,y,t) – I(x,y,t -Δ)

3) Forward MHI

Forward MHI ($H_\tau^f(x,y,t)$) is genarated a similar way with threshold is defined by :

$$\psi(x,y,t)=\begin{cases}1 & if\ D(x,y,t)\geq\xi\\ 0 & otherwise\end{cases} \tag{5}$$

with D(x,y,t) = I(x,y,t) – I(x,y,t -Δ)



a) *b)* *c)*

Fig.4: a) MHI, b) bMHI and c) fMHI of gesture Basta of depth video in CHALEARN dataset

C. Kernel descriptors on MHI

Once each video shot is represented by an MHI, we will extract kernel descriptor (KDES) from this image [2]. In the following, we will detail the step of descriptor computation. Readers could refer to [2] for more details of relevant techniques.

1) Pre-processing

As observed in the CHALEARN dataset, one hand gesture could be done by the left or the right hand, depending on the subject who realizes it. Therefore, in order to make a robust representation of the hand gesture, we do a pre-processing so that all gestures look as they are done from the same hand. Then MHI images are resized to a predefined size range and converted to grayscale ones.

2) Pixel-level features extraction

Given a normalized MHI representing one gesture, we compute the gradients at the pixels sampled on an uniform and dense grid. By doing this step, we obtain a 2-dimensional vector under the form θ(z) = [sinα cosα] representing the gradient orientation of each pixel.

3) Patch-level features extraction

A patch is defined as a square region with a predefined size around a pixel. In KDES, patch is the unit of information. The main idea of KDES is to build a metric to evaluate the similarity between two image patches. The exponential metric of Euclidean distance between pixel-level features is selected. For example considering two patches P and Q, the match kernel between their gradient features can be calculated as follows:

$$\mathrm{Kgrad}(P,Q)=\sum_{z\epsilon P}\sum_{z'\epsilon Q} m(z)m(z')k_o\big(\theta(z),\theta(z')\big)k_p(z,z') \quad (6)$$

Where:

- z, z’: denote pixels inside two corresponding patches P and Q.
- θ(z) = [sinα cosα] where α is the angle of gradient vector at the pixel z.
- m(z), m(z’): magnitudes of the gradient vectors at z, z’.
- $k_o(\theta(z),\theta(z')) = \exp(-\gamma_o\|\theta(z) - \theta(z')\|^2)$: orientation match kernel between two pixels.
- $k_p(z,z') = \exp(-\gamma p\|z - z'\|^2)$: position match kernel between two pixels.
  (Here ||a|| denotes L2-norm of vector a).

We can prove that:

$$k_o(\theta(z),\theta(z')) = [Gk_o(\theta(z),X)]^T[Gk_o(\theta(z'),X)] \quad (7)$$

Where X is a set of sampled basis vectors and G is the coefficient matrix (constructed from basis vectors).

This equation shows us an effective way to build any type of features which can be easily used for matching and results in a fast computation. We have a similar equation for position kernel. In order to calculate match kernel between two patches, each pixel of a patch needs to be matched to all the ones of the other. Hence a Kronecker product appears in the following formula showing how to compute patch-level features:

$$\mathrm{Fgrad}(P) = \sum_{z\in P} m(z)\emptyset_o(\theta(z)) \otimes \emptyset_p(z) \quad (8)$$

Where $\emptyset_o, \emptyset_p$ denote orientation and position match kernel of the pixels in a patch with the selected basis vectors (simply understood as a projection). Considering the high dimension of features vectors (due to Kronecker product), KPCA is applied with the learned eigenvectors.

We highlight the observation that using an uniform and dense grid can lead to identification errors as patches are taken even at the positions where the variance of grayscale is ignorable. In order to evaluate the importance of a patch, we propose the following metric that we call “informativity” of the patch P:

$$I(P) = \sum_{i=1}^{n} m(z_i) \quad (9)$$

Where z_i (i=1,…,n) are the pixels involved in patch P, $m(z_i)$ denotes the magnitude of the gradient vector at pixel z_i.

The larger I(P) reaches, the more informative the patch P is. We then arrange the informativities of the patches into an array IArr in the descending order. If two patches are of the same informativity, the patch appearing first in the sampling stage will be placed at the smaller index. The corresponding patch numbers are stocked in array PArr. These arrays help us eliminate a number of unimportant patches. We call Q the set of patches which are remained, Q is defined as follows:

$$Q = \{P_i \mid P_i = \mathrm{PArr}[i],\ 0 \le i \le \gamma n\} \quad (10)$$

Where P is a patch denoted by its patch number, n is the number of patches involved in the image and γ is a statistic proportion that will be selected based on the dataset.

4) Image-level features extraction

In each layer, image-level features are computed on a learned dictionary. Image-level features are extracted using spatial pyramid matching throughout a number of layers (layer 0, layer 1, layer 2, …). In layer k, an image is divided into $(2k)^2$ cells. The total number of cells generated by a division of M layers is $\frac{4^M-1}{3}$. For each cell, we first find all the patches involved in it. Each of these patches will be matched to its nearest visual word, built by the Bag of Words technique.

Then for each visual word, known as we have a list of corresponding patches, we maintain only its nearest patch. The mean value of all the distances from the patches to the visual words form the feature vector of the cell.

In conclusion, if we build a dictionary of N visual words and divide an image by M layers, then its image-level features is represented by a vector of $(N.\frac{4^{M}-1}{3})$ dimensions.

Due to our improvements on patch selection that has been discussed, if only one patch is remained for each visual word, the loss of information may happen. We therefore propose to keep 2 patches for each of these words. These patches will contribute to image-level features.

IV. Experiments

A. Dataset

The objective now is to investigate the use of MHI and gradient based KDES for hand gesture recognition. As said previously, we will evaluate our proposed method on CHALEARN challenge. This challenge focuses on the recognition of 20 Italian cultural/anthropological signs. Look inside the dataset, we found that in a hand gesture category, participants can do it in a very different manner. This dataset is therefore much more difficult than one-shot learning dataset in 2012. Although the dataset contains multimodal data, we will process only RGB and Depth data.

For evaluation since we do not have ground truth of the testing data, without loss of generality, we take a half of development dataset for training and remaining examples for testing. The development dataset is provided with 7754 video shots, each contains one hand gesture from 20 gesture categories of Italian signs.

B. Performance measures

We use two measures for recognition evaluation: Accuracy and Error rate. The accuracy is defined as follows:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \qquad (9)$$

Where TP is true positive, TN is True negative, FP is false positive and FN is false negative.

Error rate is a measure defined by CHALEARN contest. It is computed as the ratio between the sum of the Levenshtein distances from all the lines of the result compared to the corresponding lines in the ground truth value file and the total number of gestures in the ground truth value file. This error rate could exceed one.

C. Experimental results

We conduct an extensive experiments as figured in the Tab. 1. From the experiment #1 to the experiment #7, we apply on depth information, from experiment #8 to #10, we apply on color information. The experiment #11 aims to evaluate the performance of the algorithm when combining RG

B and Depth data at features level. Specifically, we concatenate the features computed from RGB and Depth data before inputting to the SVM. We try different combinations of MHI, backward MHI, forward MHI with the original KDES and the improved KDES. The results make us following conclusions:

1) MHI, fMHI and bMHI give the similar performance. Combining MHI with bMHI and fMHI on depth data make a little improvement comparing with only MHI. The combination of MHIs on color data even does not make improvements. This could be because of the redundancy in fMHI and bMHI, which might be covered in the MHI. Despite that, these results are still significantly better than simply applying KDES on MHI.

2) Normalization on hand performing gestures and the improved version of KDES make a significant improvement on the performance in both terms (accuracy and error rate).

3) Normalized MHI and improved KDES on color data give the second best performance. The reason is that the depth sensor does not give reliable information even in a near range that we call missing values in depth. Therefore, representation on Depth requires a phase to discovery depth information before using it.

4) Combining RGB and Depth data gives the best performance (experiment #11). However, it is more time consuming.

TABLE I. OBTAINED RESULTS WITH DIFFERENT EXPERIMENTS

No	Experiment	Accuracy (%)	Error rate
	Using Depth data		
1	Depth MHI + KDES	57.0	0.659
2	Normalized Depth MHI + improved KDES	60.6	0.611
3	Depth bMHI + KDES	55.8	0.672
4	Normalized Depth bMHI + improved KDES	60.7	0.604
5	Depth fMHI + KDES	54.6	0.689
6	Normalized fMHI + improved KDES	60.2	0.612
7	Normalized Depth MHI, bMHI, fMHI + improved KDES	58.28	0.640
	Using Color data		
8	Color MHI + KDES	55.7	0.664
9	**Normalized color MHI + improved KDES**	**62.4**	**0.568**
10	Normalized Color MHI, bMHI, fMHI + improved KDES	61.85	0.573
	Using both Color and Depth data		
11	Normalized (color + depth) MHI + improved KDES	**63.96**	**0.53**

Fig. 5 illustrates the recognition results on each category of hand gesture, obtained from two best trials on depth and color respectively (row 2 and row 9 of the Tab. 1). We could see that Freganiente, Fubor, Messidaccodo, Basta gestures are highly recognized. The reason is that the people perform these gestures in the similar manner, and the movement of hand is large and does not confuse with body part (Fig. 6).

Concerning other gestures, for example Vatenne or Tantotempo (see Fig. 7), the gesture has less motion and looks similar in MHI, therefore the MHI cannot represent gesture characteristic and easily confused with other gestures.

Compared to works participating to the CHALEARN contest [3], our work belongs to the middle group. The reason is that we have used only RGB and Depth information while other participants used audio video (RGB), depth and even high level features such skeleton. As reported in [5], using only audio could obtain the performance closed to ranked first in the contest due to the fact that the people could perform a hand gesture with largely difference in hand movement from other while speaking the same phase (high repeatability of audio signal). Compared to the method presented in [10], that use the keyframes extracted on depth and Multilayer Perceptron Network, our method is better. This result shows that the combination of MHI and KDES is good for hand gesture recognition.

TABLE II. COMPARISON WITH THE RESULTS OF THE CHALEARN CONTEST

Team	Score	Rank	Team	Score	Rank
Team1	0.1276	1	…	…	…
Team2	0.1539	2	Team11	0.372	11
Team3	0.1711	3	**Our best**	**0.568**	
Team4	0.1722	4	Team12	0.633	12
Team5	0.1733	5	**[10] using Depth**	**0.66**	
…	…	…	…	…	…
…	…	…	Team17	0.92	17

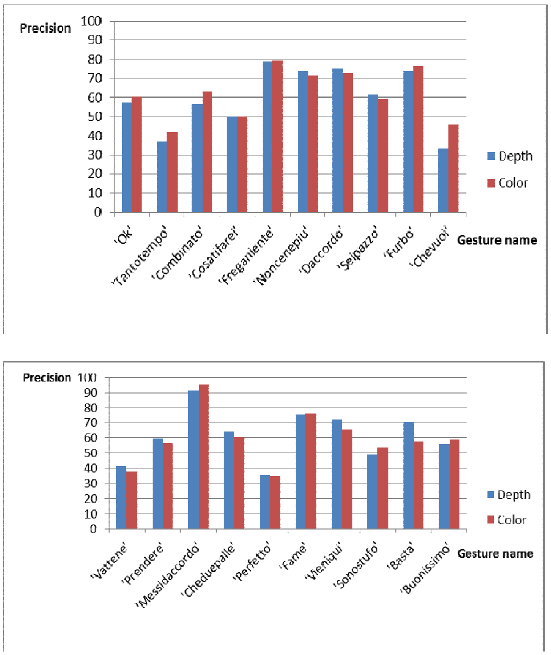


Fig. 5: Obtained accuracy for each gesture

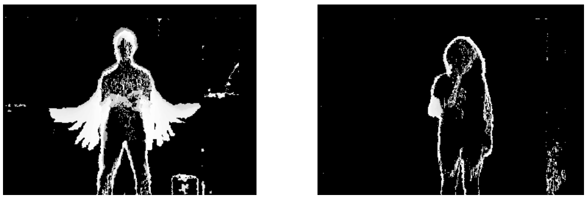


Fig. 6: a) MHI on Basta gesture; b) MHI on Furbo gesture

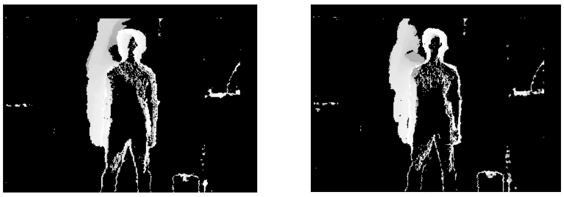


Fig. 7: a) MHI on Vatenne gesture; b) MHI on Tantotempo gesture

V. Conclusions

This paper presented a new method on dynamic hand gesture recognition. The proposed method represents movement of gesture by motion history image and extracts kernel descriptor from this image. Finally, SVM have been used for hand gesture classification. We have conducted an extensive investigation on different types of MHI as well as their combination to make more informative representation of the gesture motion. In addition, we have made two improvements for KDES extraction step. The method has been evaluated on challenging dataset and shows how MHI and KDES could contribute for hand gesture recognition. Currently, our method belongs to the middle group. The reason is that we have used only Depth information. In the future we will combine this descriptor with other features extracted from audio, skeletal data to improve the performance.

Acknowledgment

This research is funded by Hanoi University of Science and Technology under grant number T2014-100.

References

1. Bobick, A.F. and J.W. Davis, *The recognition of human movement using temporal templates.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001. **23**(3): p. 257-267.

2. Bo, L., X. Ren, and D. Fox, *Kernel Descriptors for Visual Recognition*, in *Advances in Neural Information Processing Systems (NIPS)*2010.

3. S. Escalera, J.G., X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athistos, H.J. Escalante,, *Multi-modal Gesture Recognition Challenge 2013: Dataset and Results.* ICMI workshop, 2013.

4. S. Mitra , T.A., *Gesture Recognition: A Survey.* IEEE Transactions on Systems, Man, and Cybernetics, 2007. **37**(3): p. 311-324.

5. J. Wu, J Cheng, C. Zhao, H. Lu, *Fusing Multi-modal Features for Gesture Recognition*, in *ICMI workshop* 2013: Sydney, Australia.

6. I. Bayer, T.S., *A Multi Modal Approach to Gesture Recognition from Audio and Video Data*, in *ICMI workshop*2013: Sydney, Australia.

7. X. Chen, M.K., *Online RGB-D Gesture Recognition with Extreme Learning Machines*, in *ICMI workshop*2013: Sydney, Australia.

8. K. Nandakumar et al., *A Multi-modal Gesture Recognition System Using Audio, Video, and Skeletal Joint Data*, in *ICMI workshop*2013: Australia.

9. B. Ni, G.W., P. Moulin, , *RGBD-HuDaAct: A Color-Depth Video Database For Human Daily Activity Recognition*, International Conference on Computer Vision Workshops (ICCV Workshops), 2011: p. 1147 - 1153.

10. N. Neverova, C.W., G. Paci, G. Sommavilla, *A multi-scale approach to gesture detection and recognition*, in *ICCV Workshop on Understanding Human Activities: Context and Interactions (HACI 2013), Sydney, Australia.*2013. p. p. 484-491