Tremor Detection Using Motion Filtering and SVM

Bilge Soran* Jenq-Neng Hwang‡ Su-In Lee* Linda Shapiro*‡
*Dept. of Computer Science and Engineering ‡Dept. of Electrical Engineering
University of Washington, Seattle, WA 98195, USA
{bilge@cs, hwang@u, suinlee@cs, shapiro@cs}.washington.edu

Abstract

The hand tremor is one of the most common motion disorders caused by various neurological diseases. Currently diagnostic procedures for tremor evaluation are subjective, and there are no examinations available that can accurately indicate whether tremors are present in a patient's daily life. Early detection of tremor is extremely important for the cure of the disease that causes the tremor. Thus, in this study we aim to develop a computational method based on machine learning that can automatically detect hand tremors from a video of a patient when the patient is doing his/her daily activities. The main challenge in tremor detection is that motion is very subtle, and the signature of the motion can vary from patient to patient. We first generate a training data set consisting of 173 simulated tremor/non-tremor video files. They contain very subtle and less discriminative motions. The main contributions of our study are the personalized skin detection, motion filtering and feature extraction pipeline. We evaluated our method through leave-one-out cross validation (LOOCV) testing and showed that in preliminary tests our method achieved 95.4% recognition accuracy. [1]

1. Introduction

The use of computer vision in surveillance, monitoring and medical imaging has matured for years; with the advances in the hardware many time-consuming algorithms can be run in real time. The main motivation in this project is to produce a systematic methodology that can run in real time, adapt the system to users, and assist doctors to decide on diagnoses. These types of systems are especially useful when there is no other objective measurement available. We show in this pilot study that with a simulated tremor/non-tremor dataset, our prediction system is fairly accurate and fast enough to adapt to real time.

Section 2 reviews some related work and provides the motivation for this research. The proposed methodology is described in Section 3. Section 4 shows the experimental results, followed by the conclusion in Section 5.

2. Related Work

The purpose of human action recognition systems is to automatically analyze the actions of the people from videos. The ability to recognize human actions and gestures is important for surveillance systems, rehabilitation purposes, diagnosing symptoms of a particular illness, home monitoring, intelligent robotics, human-computer interaction, etc. Among these, hand gestures recognition systems are especially important for virtual reality, robotics, games, smart surveillance, sign language translation and medical systems. Garg et al. [11] categorize the existing hand gesture recognition approaches into either 3D model-based approaches or appearance-based approaches. A straightforward method used in appearance-based approaches is to extract skin-colored regions from the image. Another method is to use the eigenspace to produce a compact description of a large set of high-dimensional data using a small set of eigenvectors [11]. Murthy and Jadon [10] adopted an appearance-based approach based on low-level hand features from a collection of 2D intensity images. However, if a system is able to extract appropriate features that describe characteristics of each action's 3-D (X, Y, T) volumes, the action can be recognized by solving an object matching problem [2].

Similar to ours, there are other studies targeted at healthcare. Cuppens et al. [8] developed an algorithm, based on the Horn-Schunck optical flow, to detect movement epochs from video in nocturnal datasets for pediatric epileptic patients. Ghali et al. tried to inte-

[1] The authors thank Xerox Corporation for a gift that partially supported this research.

grate virtual reality and machine vision technologies to produce innovative stroke rehabilitation methods. They proposed a combined object recognition and event detection system that provides real time feedback to stroke patients performing everyday kitchen tasks necessary for independent living [1]. The most similar work to ours was done by Uhrikova et al. [12], who also worked on hand tremors, but their purpose was different from ours in that they tried to measure the hand tremor frequencies instead of classifying tremor instances from non-tremors. They compared their calculated frequencies with those measured by an accelerometer.

3. Methodology

As shown in Figure 1, our proposed method consists of three algorithmic stages. The first one performs the (personalized) skin detection from the video of a person to locate the hand skin blobs frame-by-frame. Based on the segmented hand skin blobs, the second stage extracts the frequencies of the temporal motion change patterns as our features for tremor detection. Finally, the third stage distinguishes tremor instances from non-tremor instances. To perform skin detection, a personalized skin model is generated from one video of a person. This model is used to extract the skin blobs from training images; therefore the hand skin color in the training images should have (or have been calibrated with) similar skin color distribution with the dataset used for training the classifier and the subsequent testing videos. After skin detection, a blob extraction method is applied to extract the connected skin blobs corresponding to the hand locations (since the videos in the dataset contain only hand regions, as shown in Figure 2(a), additional hand detection is not needed). After hand blobs are extracted, directional motion features and the corresponding temporal changes can be obtained based on the optical flow of the pixels. These motion features are further converted, based on Discrete Cosine Transform (DCT), to extract the frequency of the directional motion changes. Finally, a classifier is trained using a Support Vector Machine (SVM) with the Radial Basis Function (RBF) kernel.

Figure 1: Three algorithmic stages of our proposed hand tremor detection system.

The skin detection and blob extraction process is described in Section 3.1. The feature extraction process is explained in Section 3.2, and classification is discussed in Section 3.3.

3.1 Personalized Skin Detection

Our skin detection method is inspired by the adaptive skin detection method in [3]. For the training/testing set used in our experiments, a simple threshold-based skin detection would be enough, since the hand motions are recorded on a black background. However, since the ultimate objective of this research is to develop a system that detects hand tremors of a person while he/she is doing his/her daily activities in real time in a normal environment, and since the skin models are usually not generic enough to work for everyone, we designed a personalized skin detection system. In order to train this skin detector model, the end-user is asked to move his/her hand for 3 seconds to obtain a training video. The skin detector decides on the skin regions by both thresholding and using the moving pixels of the video as described below.

For personalized skin modeling only 3 seconds of one 30 fps video (90 frames, converted from RGB to HSV format) of one person is used. For each frame of the training video, a hue threshold and an intensity threshold are applied to select probable skin pixels on hue and intensity planes to obtain thresholded hue and intensity planes (HueT, IntensityT). Then a median filter is applied on the HueT and IntensityT to remove salt and pepper noise on both planes, and the intersection of the filtered planes is computed to obtain a probable skin region (ProbableSkin). The dense optical flow [9] is then computed on the ProbableSkin region, and the pixels having a larger optical flow magnitude are selected (MovingPixels). By intersecting ProbableSkin and MovingPixels regions, a FinalMerged image is obtained, and two histograms corresponding to the hue and intensity colors of the FinalMerged image are derived. Finally, a Gaussian function is fit to each histogram and used as the skin model for further processing. After selecting the pixels according to the generated skin model, a standard blob detection algorithm for extracting 8-connected components [5] is applied, and regions having a size smaller than a pre-specified threshold are discarded.

3.2 Feature Extraction

The feature extraction stage starts with applying the Lucas-Kanade [9] optical flow (OF) detector for each blob in three consecutive frames. Because our goal is to detect subtle motions, we need to calculate dense optical flow, which is very computationally expensive if calculated on every pixel. Thus one of every 4 pixels of the blob area is used. Then the selected pixels' locations on the subsequent three frames are used to calculate the

motion directional changes. Figure 2(a) shows the positions of the selected points in these three consecutive frames, with a sampling rate of 10 for a clearer representation.

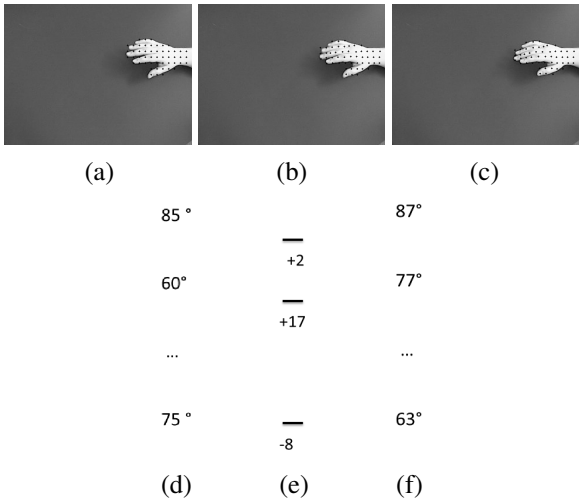

Figure 2: (a) Initial points. (b) Positions on the next frame. (c) Positions after two frames. (d) Motion angles from frame (a) to (b). (e) MDC from frame (a) to (c). (f) Motion angles from frame (b) to (c).

In a tremor movement, the main discriminative features are not the motion directions themselves, but the motion directional *changes* between consecutive frames. While directional features are sensitive to rotation, directional change features are more rotation, scale and translation invariant. Therefore, to extract the tremor signature patterns, motion directional changes (MDC) are calculated. To extract the angle of the MDC (see Figure 2 (e)) for each hand blob, we subtract the motion directional angles (see Figure 2 (d) (f)) between 3 consecutive frames. To have more precise hand location information, instead of using the position inferred from optical flow and suffering from error propagation, the process of calculating MDCs should use the skin blobs, separately derived with skin detection and blob extraction from every frame.

Note that while one part of the hand can move upward, another part can move downward, and a tremor can exist in any region of a hand. We assume (and have observed through experiments) that if a tremor occurs, the directional change of the tremor dominates all directional changes that can exist in any region of the hand. Moreover, not all hand regions exhibit tremor behavior, and averaging the MDC features over the whole hand can smooth the values while decreasing the discriminative power of the features. Therefore, we need to identify the hand regions that exhibit significant MDCs related to the tremor. For this purpose, the MDC values in the hand blob are sorted from the most positive to the most negative. If the sum of the MDC in the hand blob is positive, we take the average of top P% of the sorted values as our representative MDC; otherwise, if it is negative, we take the average of the bottom P% of the sorted values. The result of this process gives us the representative feature component of one frame. The procedure is repeated until all the frames in the video are processed, to get a 118- (out of 120 frames) dimensional feature vector, which represents the temporal evolution of dominating motion directional changes. To detect tremor, the frequency of the MDC is more useful. Therefore, we apply the Discrete Cosine Transform (DCT), as defined in Equation 1, to the feature vectors. Note that, we explored other feature settings, like a combination of motion direction and magnitudes, motion directions or MDC but not in the frequency domain, and different percentages of MDC features from hand blobs, but the DCT applied to MDC features with an MDL discretization [6] produced the most discriminative features among them (see Section 4).

$$y(k) = w(k) \sum_{n=1}^{N} x(n) cos(\frac{\pi(2n-1)(k-1)}{2N}) \quad (1)$$

$$k = 1, 2, ...N \text{ and } w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases}$$

Here, $N = 118$ is the length of the feature vector x, and the resulting DCT vector y has the same size.

3.3 Classification

A classifier, which can distinguish tremor instances from non-tremor instances, is developed using an SVM [4] with Gaussian RBF kernel. When using an SVM, training vectors (N-dimensional DCT features) are mapped into a higher (maybe infinite) dimensional space by the kernels. The SVM finds a linearly separable hyper-plane with the maximal margin in the higher dimensional space. The RBF kernel has less parameters than a polynomial kernel and helps to improve the mapping accuracy, especially when the relationship between class labels and attributes are nonlinear [7]. Since the number of features is not very large, and the data is not linearly separable, RBF is chosen as the kernel. Its definition is given in Equation 2.

$$exp(-\gamma * |u - v|^2) \quad (2)$$

where $\gamma = \frac{1}{2\sigma^2}$ of a Gaussian with variance σ. In our experiments we used a γ of 0.1.

Besides SVM we also experimented with other classifiers such as Random Forests, Naive Bayes and Logistic Regression. Among all, the SVM with RBF achieved either comparable or the best accuracy.

4. Experiments

We experimented with three cross-validation settings with different top/bottom P% of points selected from hand blobs. As can be seen from Figure 3, our experiments showed that taking the top/bottom 25% of the MDC of the hand blob to describe tremor features gave the best results among all three settings. In the first setting, the classifier model was tested using a 5-fold cross-validation; then a 10-fold cross-validation and LOOCV were tried. Moreover, to improve the classification performance and to remove the noise, the minimum description length (MDL) discretization method, proposed by Fayyad and Irani [6], was applied to the feature vectors. A drastic improvement in the classification accuracy of the trained SVM is observed when MDL discretization is applied. The MDL discretization approach recursively partitions all the known values of a feature into subintervals until minimal description length is achieved [6]. Figure 3 summarizes the results for the features selected from top/bottom 10% and 25% of the MDC features of the hand blob.

% of MDC features from hand region	5 fold Cross-Validation	10 fold Cross-Validation	Leave One-Out Cross Validation
10 %	80.3 %	80.3 %	79.2 %
10 % with MDL discretization	90.8 %	90.8 %	89.6 %
25 %	85.0 %	86.1 %	86.7%
25 % with MDL discretization	94.8 %	94.8 %	**95.4 %**

Figure 3: Classification accuracy

For this research, a tremor/non-tremor dataset consisting of 90 simulated tremor cases and 83 normal hand movement cases were recorded. Each video was recorded by a fixed network camera with static background in 30 fps in the indoor environment with 6 human subjects. They have a resolution of 320×240 and a duration of at least 4 seconds (120 frames). Some of the simulated tremors are extremely subtle, and some of the non-tremor videos contain movements very similar to tremor cases for the challenge. Since our final goal is to build a real time monitoring system that distinguishes hand tremors, we believe such a system should be able to distinguish these challenging cases from the tremor cases and catch the very subtle tremors at the same time, like tapping your fingers versus tremor.

5. Conclusion

The hand tremor is one of the most common motion disorders and can be a sign of certain neurological diseases. In this study, we built a system that can automatically distinguish hand tremors from other kinds of hand movements. The proposed method requires only a video of a person, which makes it suitable for diagnosis purposes. It has the capability to classify very subtle motions and distinguish tremor-like movements, such as tapping fingers or waving hand very quickly, from tremor movements. In this research, a dataset of simulated tremor/non-tremor hand motions is prepared and the methodology tested on this dataset. Although tremor detection is a hard problem, in this pilot study, with the described methodology a classification accuracy of 95.4% is achieved with LOOCV. We believe the highly accurate segmentation of the videos has also contributed to this high accuracy. In our future studies, we are going to apply this methodology to videos of people from more diverse ethnicity, with ordinary backgrounds and in real time.

References

[1] A. Ghali, et al. Object and event recognition for stroke rehabilitation. In *VCIP*, 2003.

[2] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 2011.

[3] G. Bradski. The OpenCV Library, 2000.

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2011.

[5] F. Chang, C.-J. Chen, and C.-J. Lu. A linear-time component-labeling algorithm using contour tracing technique. *Comput. Vis. Image Underst.*, 2004.

[6] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous valued attributes for classification learning. In *13th IJCAI*, 1993.

[7] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A Practical Guide to Support Vector Classification, 2000.

[8] K. Cuppens, et al. Detection of epileptic seizures using video data. In *Proc of the 6th Intern Conf on IE*, 2010.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th IJCAI*, 1981.

[10] G. Murthy and R. Jadon. A review of vision based hand gestures recognition. *IJITKM*, 2009.

[11] P. Garg, et al. Vision based hand gesture recognition. *Engineering and Technology*, 2009.

[12] Z. Uhrikova, et al. Action tremor analysis from ordinary video sequence. *Conf Proc IEEE Eng Med Biol Soc*, 2009.