Chapter 40
Using Time Proportionate Intensity Images with Non-linear Classifiers for Hand Gesture Recognition

Omar Ahmad, Basilio Bona, Muhammad Latif Anjum and Ikramullah Khosa

Abstract Gestures are signals that contain important spatiotemporal information. Understanding gestures is a trivial task for humans, but for machines it is a challenging task involving thousands of computations per video frame. This paper investigates an efficient hand gesture recognition technique which is based on time projections of the hand location. For recognition, non-linear classifiers, namely Support Vector Machines and Artificial Neural Networks, are tested. The proposed method performs much faster than the conventional Markov Model based gesture recognition techniques while achieving comparable recognition results.

Keywords Gesture recognition · Spatiotemporal segmentation · Computer vision · Machine learning · Classification · Human robot interaction

40.1 Introduction

Gestures are parts of human movement which contain certain information. They can convey certain meanings (like in Sign Language), or commands, or can be used to point to certain objects in the surroundings. From a computational point of

O. Ahmad (✉) · M. L. Anjum
Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129 Turin, Italy
e-mail: omar.ahmad81@gmail.com

B. Bona
Department of Control and Computer Engineering (DAUIN), Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129 Turin, Italy

I. Khosa
Department of Electronics and Telecommunications (DET), Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129 Turin, Italy
e-mail: ikramullahkhosa@gmail.com

H. A. Mat Sakim and M. T. Mustaffa (eds.), *The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications*,
Lecture Notes in Electrical Engineering 291, DOI: 10.1007/978-981-4585-42-2_40,

view they can be thought of as spatiotemporal signals that contain certain information which is vital for robots or machines to interact with their environments. A lot of research has been carried out in the recent years to develop techniques for gesture recognition [1]. From computer gaming to touch pads, and other such devices, we see a lot of applications of gesture recognition as results of this research.

As the computational power of the machines is rising with advances in IC manufacturing technology we see a lot of research activity in the field of Human Robot Interaction (HRI). The dream of making robots a household commodity is suddenly looking very much realizable in near future. Recent research is aimed at making the human-robot interaction natural and with as fewer constraints as possible. Humans interact with each other naturally, robustly identifying gestures as well. The robots have come a long way in understanding their environment, being now able to see, hear and feel to a certain limited extent, but a lot more is yet to be achieved. Understanding gestures robustly and naturally is a task that still remains an open area for research.

In this paper we present a gesture recognition technique that can robustly detect gestures without requiring any temporal segmentation beforehand. The proposed method performs the higher level tasks of gesture spotting, i.e., determining the start and end frames of the gestures (temporal segmentation) and classification, as well as low level tasks of spatial segmentation of the hand at each frame.

The algorithm starts by spatially segmenting the hand for every incoming frame by combining skin detection and motion calculation. This information is then sent to the Time Proportionate Intensity Accumulator unit which stores this information as an intensity image (TPI image). A high intensity on the TPI image corresponds to a recent impression and lower intensities correspond to previous impressions. Information from each new frame, after preprocessing is sent to the classifier unit which in turn classifies whether or not a gesture has just been performed. If a gesture is performed it is duly classified as being one of the gestures in the system vocabulary. Figure 40.1 is a block level representation of the proposed gesture recognition system.

A key aspect of the proposed approach is its very low computational cost. The algorithm can classify the gestures in real time mainly because it does not model the gestures as Markov Chains. Thus it does not have to match a Hidden Markov Model with a video query comprising of a series of frames.

The rest of the paper is organized as follows. Section 40.2 describes the recent research work in gesture recognition and how it relates to our work. Section 40.3 discusses the preprocessing step of spatial segmentation of hand and temporal projection. Section 40.4 is about the nonlinear classifiers used for gesture recognition and their offline training. In Sect. 40.5 we discuss the experimental setup, the video sets used and the results of classification, concluding with recommendations and remarks on future work in Sect. 40.6.

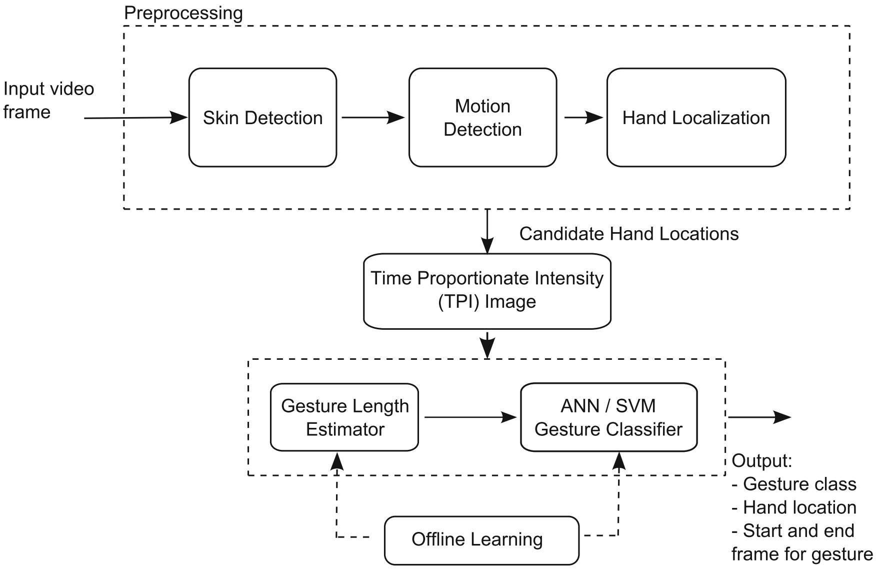

Fig. 40.1 Block diagram of the proposed gesture recognition system

40.2 Related Work

An important discriminating feature of the proposed algorithm from existing gesture recognition algorithms is that it does not require temporal segmentation of the gesture as preprocessing like other dynamic approaches [2–5]. In many gesture recognition algorithms [6, 7], spatial and temporal segmentation is done at a lower level and certain features of shape and velocity are extracted as preprocessing steps. These features are then passed to the classifier for recognition [6]. Recognition results deteriorate if these preprocessing segmentations fail. Although we perform a preprocessing step involving spatial segmentation of the hand, its failure in some frames does not make the next stages of the algorithm to fail. Ambiguities in hand segmentation causes some noise in the background in our algorithm but the nonlinear classifiers can handle such noise.

Pavlovic et al. [8], discuss the template matching based segmentation of the hand useful in determining the posture of the hand. They trained a hand shape determining classifier to detect hand posture and used skin color based classifier to spatially segment the hand. Aaron and James [9], discuss temporal templates for learning the history of motion. The proposed method is similar in the sense that it also captures the history of motion onto a spatial plane but differs in linear fading concept that we introduce.

Some algorithms like [3, 10, 11], extract global features from each frame like motion fields or use a transformed set of images such as intensity thresholded images or difference images as inputs to gesture recognition modules. These algorithms do not incorporate tolerance to background movements which can cause these algorithms to fail in noisy environments. We train our nonlinear classifiers for noisy environments as well thus it outperforms these other algorithms.

Most of the gesture recognition algorithms model the gestures as Markov Chains [12–14], with fixed or variable transition probabilities. The recognition of gestures in these algorithms becomes a problem of state by state aligning the query video sequence with the gesture model. This involves computations that increase exponentially with the gesture models in the vocabulary. To overcome this issue of time complexity these algorithms devise certain mechanisms to prune out certain hypotheses and rely on Dynamic Programming (DP) to reduce the computations. As we do not model gestures as Markov Chains our algorithm has to perform significantly less number of computations for classification.

The proposed method is similar to algorithms like [11] and [15], which model the gestures as rigid 3D patterns. These algorithms do not perform well if the gesturing speed is changed. On the other hand we learn the tolerances in gesturing speed from the training data and do not face these problems.

Finding the start and end frames in a gesture is referred to as *gesture spotting*. Algorithms can be divided in two categories based on the mechanisms they adopt for gesture spotting. One approach is to temporally segment the gestures before classification. This is usually achieved by inserting intervals between the gestures [16, 17]. The other approach indirectly performs the task of gesture segmentation based on results of certain cost functions over a window in time which slides over the temporal axis of the incoming video stream [7, 18, 19]. Our approach also lies in this second category where we find the start and end frames of a gesture during the classification.

Then there is the problem of sub gestures, i.e., when some gestures are part of other longer gestures. Classifiers usually either do not consider this possibility by imposing limitations on the gestures themselves [19] or they require additional looping over all the gestures in the vocabulary to determine these sub gestures. The proposed algorithm in its output not only classifies a gesture it also gives a confidence measure and expectancy of a super gesture.

40.3 Preprocessing and Hand Segmentation

In this section we describe the preprocessing steps performed on each frame before it is passed on to the classifier stage. First the spatial segmentation techniques to get the most probable hand locations are presented. Next discussed are the methods to incorporate multiple hand candidates and background noise removal. At the end of this section we describe in detail, the working of Time Proportionate Intensity Accumulator unit to get the projections of the temporal axis onto the spatial plain (TPI image) which in turn allows us to use existing techniques of feature based classifiers to recognize the gestures.

40.3.1 Hand Segmentation Based on Skin Pixel Estimation

First of all a skin likelihood image is computed using the point operation of Eq. (40.1). The mean μ_s, and variance $\sum_S$ from a generic skin model of [20] are used. Next the motion mask is calculated by taking the difference of the current frame and the previous frame. Hand likelihood image is obtained by applying the motion mask to the skin likelihood image.

$$p(x|skin\ pixel) = \frac{1}{(2\pi)^{\frac{1}{2}}|\sum_s|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x-\mu_s)^T \sum_s^{-1}(x-\mu_s)\right) \tag{40.1}$$

The hand likelihood image is filtered with a 3 × 3 order statistic median filter to remove background noise. This hand likelihood image is then passed onto the next stage of k-means clustering.

40.3.2 Hand Localization Using k-mean Clustering

This module takes as input the binary hand likelihood image. The white pixels in this image correspond to a high likelihood of hand location. The indices of all non-zero pixels are extracted from this image and are used to find K clusters. Each cluster corresponding to high probability of hand locations. The number of member points for each cluster determines the size of the impression made by this hand hypothesis on the TPI image in the next stage. This is approach is similar to the one used in [21] except that instead of using an integral image, and moving window, k-mean algorithm is used for more accurate hand localization.

The algorithm can be extended to accommodate two handed gestures for example and to make the algorithm robust. This also helps reduce the background noise due to moving distracters in the background. Figure 40.2 shows the results of outlier rejected hand segmentation.

40.3.3 Time Proportionate Intensity Projections

We get K candidate hand locations from the clustering unit. We place K blobs on the Time Proportionate Intensity Accumulator at the spatial locations received from the previous unit. We assign them maximum intensity. Upon processing of the next frame we receive further K blobs to be placed on the accumulator. Here we decrement the accumulator first by the fading factor α before placing the blobs. This way the TPI image periodically forgets or fades away the impressions made by previous frames, Fig. 40.3.

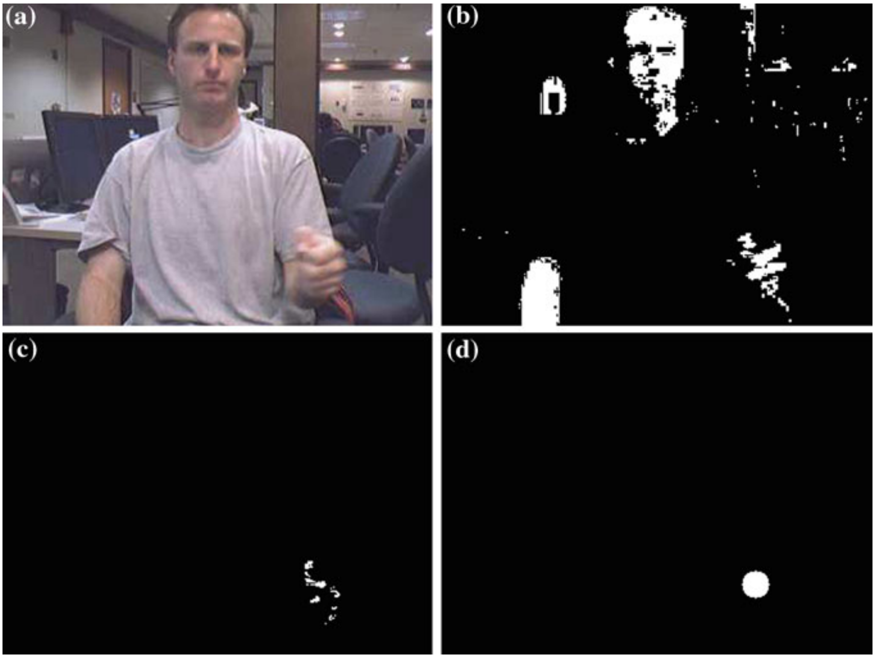

Fig. 40.2 Preprocessing of an incoming frame and corresponding impression onto the time proportionate intensity accumulator

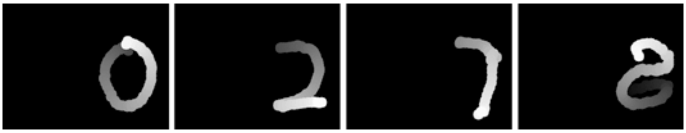

Fig. 40.3 Example time proportionate intensity (TPI) accumulators from training sample videos (K = 1)

We have used a linear fade where the intensity of a pixel fades in the accumulator by a constant α after each new frame.

$$I_t = I_{t-1} - \alpha \tag{40.2}$$

To get training data, TPI image is thresholded (Eq. 40.3) to zero to forget all the information $\boldsymbol{n}$ frames ago thus temporally segmenting the gesture. The gesture length for each gesture in the training set videos is available in the ground truth files. Average gesture length is also learned from the training data for each gesture class.

$$threshold = max(I) - n\alpha \tag{40.3}$$

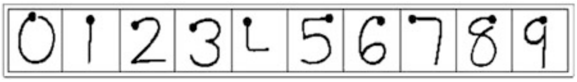

Fig. 40.4 Palm's Graffiti digits

40.4 Spatiotemporal Matching

In this section the implementation of our gesture class learning method using nonlinear classifiers (Neural Networks and Support Vector Machines) is presented first, and then we discuss the classification mechanism used.

40.4.1 Model Learning and Non Linear Classifiers

We have used the video data sets by Athitsos [18] for experimentation. These video sets include a training set, an easy data set and a difficult data set. Each set contains 30 videos where signers make signs of the Palm's graffiti digits 0 to 9, Fig. 40.4. In each set of videos there are a total of 10 different signers with 3 videos from each signer. The signers wear colored gloves in the training data set only. All the videos are accompanied by ground truth text files which contain the temporal segmentation of the gestures. Meaning that start and end frames of all the gestures in each video are given. These data are useful for training and cross validation purposes.

40.4.2 Classification Using Support Vector Machines

Model learning phase includes feeding the TPI image state just after the gesture is completed to the SVM trainer. The TPI image is thresholded to forget all the information before the start of the current gesture. The m × n TPI image is down sampled to 25 × 25 image and then reshaped into 1 × 625 feature vector that along with the gesture label is used to train the SVM parameter vector theta θ. The optimization objective function for SVMs is a minimization problem given in Eq. (40.4). The kernel function we have used is the Gaussian kernel of Eq. (40.5).

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1\left(\theta^T f^{(i)}\right) + \left(1 - y^{(i)}\right) cost_0\left(\theta^T f^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2 \tag{40.4}$$

$$f^{(i)} = k\left(x^{(i)}, l^{(i)}\right) = exp\left(-\gamma \|x^{(i)} - l^{(i)}\|^2\right), \quad \gamma = \frac{1}{2\sigma^2}, \ and\ \gamma > 0 \tag{40.5}$$

The training data is divided into 11 classes. 10 classes for gestures 0 to 9 and one class for training examples where no gesture has been performed. Incomplete gestures and partially observed gestures are placed in this class. For the implementation of the SVMs we have used the library package LIBSVM [22].

For each incoming frame we get the current state of the TPI image as described in the previous sections. A window with this TPI image is passed to the classifier after each frame.

We use SVMs trained as multi-class classifiers. LIBSVM implements the multi-class classifiers using one against one approach. Therefore this TPI image is matched against all the models in the vocabulary. A positive match (based on majority voting) gives the gesture class as well as the temporal segmentation of the gesture as we now know the end frame of the gesture. The average time (in number of frames) for that particular gesture class was already learnt in the model learning stage.

Temporal segmentation can be improved if after initial classification we move the time window and recalculate the cost finding the frame corresponding to the minimum cost as the start frame. Of course this will be under the hypothesis that the cost function is convex with a single minimum.

40.4.3 Classification Using Artificial Neural Networks

Artificial Neural Network (ANN) is a computing system, composed of large number of highly interconnected units (called neurons) that emulate the organization and operation of biological nervous system. This is one of the most widely used technique in classification and pattern recognition problems [23].

For the learning of ANN, special training algorithms are developed based on the learning rules similar to learning mechanisms of biological systems. There are many types and architectures of neural networks, fundamentally depending on their learning mechanisms. For gesture recognition, we have chosen a three layer Multilayer Perceptron Neural Network (MLPNN) with back propagation training algorithm, consisting of one input, hidden and output layer each. Architecture of a typical MLPNN with three layers is shown in Fig. 40.5. The input layer of our network consists of 625 features and output layer has ten neuron; equal to the number of target classes. The number of nodes in the hidden layer has great influence on the performance of the network. An optimum number of nodes in the hidden layer are selected after trial and error in network performance. For the implementation of the ANNs we have used the OpenCV CvANN Class. For back propagation based training, the library uses the algorithm proposed in [24].

40.5 Experiments and Results

This section explains the experimental setup. Specifically we discuss the choice of data set of videos, the number of training examples, the validation data set and the test sets. Here we also present the results of classification and compare them with existing methods.

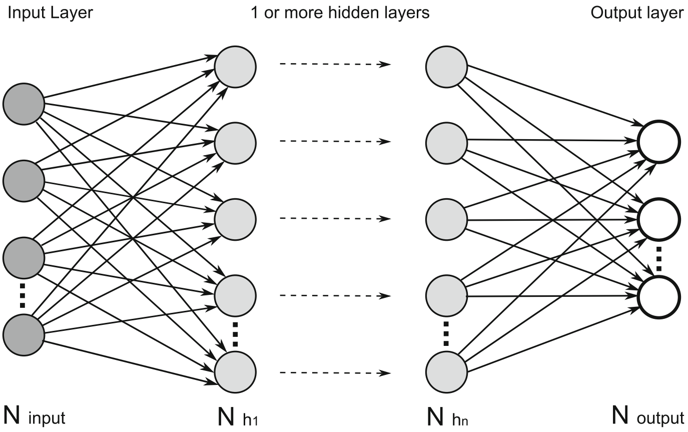

Fig. 40.5 Architecture of a multi-layered artificial neural network

40.5.1 Results Using Support Vector Machine

The training data set has only 30 videos, 3 from each of the 10 signers. Extracting a single TPI image per gesture class from every video leads to only 30 samples for training purposes. These are very few when training the SVMs to obtain good results. One way of increasing the samples is to extract multiple TPI images per gesture class from each video. This way multiple TPI images from above 80 % gesture completion can be extracted to increase the training examples. Following tables present the early stage results of the experimentation. As seen in Table 40.1, the accuracy of classification for the difficult data set is very poor. This is due to the background motion in the difficult data set. We are currently working to devise methods to incorporate multiple hand hypotheses with low time complexity.

40.5.2 Results Using Artificial Neural Networks

We have a total of 4,634 samples of hand gestures corresponding to ten target classes; 0–9. Sixty percent of the data representing each class is used for training, twenty percent is used for cross validation (CV) and the rest twenty percent is considered as test data. The network is trained with back propagation algorithm and a regularization parameter is also tuned to overcome the problem of over fitting as well as under fitting of data. The number of epochs is limited to 100. Using the best results of network on CV data, an optimum architecture of the network is settled with 196 nodes in the hidden layer. The network is then used to classify the test data. The overall network accuracy is presented in Table 40.2, and confusion matrix for test data is presented in Table 40.3.

Table 40.1 Class wise accuracy and false positives for easy (E) and difficult (D) data sets

Gesture	Accuracy percent		False positives		Confused with	
	E	D	E	D	E	D
0	74	34	5	19	6	6
1	81	73	3	6	7	9
2	91	56	2	14	4	7
3	78	43	4	15	2	2
4	85	67	3	9	2	2
5	81	48	3	13	8	8
6	88	62	2	10	0	0
7	87	69	2	9	1	2
8	76	33	5	20	5	5
9	77	35	5	19	7	1

Last column shows class label a gesture is confused with the most

Table 40.2 Accuracy accross the training, cross validation and the test data sets

Number of hidden layers	Training accuracy (%)	Cross validation accuracy (%)	Test accuracy (%)	Regularization parameter (λ)
182	99.89	92.12	96.24	0.005

Table 40.3 The confusion matrix for ANN

Actual class	Predicted class									
	1	2	3	4	5	6	7	8	9	10
1	102	0	0	0	0	0	0	0	0	0
2	0	107	0	0	0	0	0	0	0	0
3	0	0	78	0	14	0	0	0	0	0
4	0	0	0	82	0	0	0	0	0	0
5	0	0	18	0	78	0	0	0	0	0
6	0	0	0	0	0	95	0	0	0	0
7	0	0	0	0	0	0	80	0	0	0
8	0	0	0	0	0	0	0	90	0	0
9	0	0	0	0	0	0	0	0	99	0
10	0	0	0	0	0	0	0	0	0	86

40.6 Future Work and Conclusions

We presented a method of recognizing gestures with minimal computations thus making run-time gesture classification possible. Currently the algorithm is designed to recognize one-handed gestures. It can be modified to be able to accommodate two handed gestures or even other classes of gestures. Detection of right and left hands using the existing techniques of Viola and Jones [25] is possible. In this way two handed gestures can be recognized in run-time without having to compromise the speed.

One major drawback of the algorithm is the drop in accuracy of classification with movements in the background (tested on difficult data set). This is mainly due to the implementation decision of using one single reliable hand location, for impression on the TPI image. Although we do train our classifier with the background noise, thus increasing the accuracy in recognition other techniques need to be used to heuristically minimize the noise.

Hence the algorithm is suitable for time critical applications or for slower systems, which are incapable of running the traditional Markov Chains based algorithms successfully, and with fairly minimal background movements.

References

1. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990. ISSN 0262-8856, doi:10.1016/j.imavis.2009.11.014
2. Corradini A (2001) Dynamic time warping for off-line recognition of a small gesture vocabulary. In: Proceedings of IEEE ICCV workshop recognition, analysis, and tracking of faces and gestures in real-time systems, pp 82–89
3. Darrell T, Pentland A (1993) Space-time gestures. In: Proceedings of IEEE conference computer vision and pattern recognition, pp 335–340
4. Cutler R, Turk M (1998) View-based interpretation of real-time optical flow for gesture recognition. In: Proceedings of third IEEE international conference on automatic face and gesture recognition, pp 416–421
5. Starner T, Weaver J, Pentland A (1998) Real-time American sign language recognition using desk and wearable computer based video. IEEE Trans Pattern Anal Mach Intell 20(12):1371–1375
6. Yang MH, Ahuja N, Tabb M (2002) Extraction of 2D motion trajectories and its application to hand gesture recognition. IEEE Trans Pattern Anal Mach Intell 24(8):1061–1074
7. Oka K, Sato Y, Koike H (2002) Real-time fingertip tracking and gesture recognition. IEEE Comput Graphics Appl 22(6):64–71
8. Pavlovic V, Sharma R, Huang T (1997) Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans Pattern Anal Mach Intell 19(7):677–695
9. Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. IEEE Trans Pattern Anal Mach Intell (PAMI) 23(3):257267
10. Quattoni A, Wang S, Morency L-P, Collins M, Darrell T (2007) Hidden conditional random fields. IEEE Trans Pattern Anal Mach Intell 29(10):1848–1852
11. Gorelick L, Blank M, Shechtman E, Irani M, Basri R (2007) Actions as space-time shapes. IEEE Trans Pattern Anal Mach Intell 29(12):2247–2253
12. Chen F, Fu C, Huang C (2003) Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Video Comput 21(8):745–758
13. Sato Y, Kobayashi T (2002) Extension of hidden Markov models to deal with multiple candidates of observations and its application to mobile-robot-oriented gesture recognition. In: Proceedings of 16th international conference on pattern recognition, vol II. pp 515–519
14. Alon J, Athitsos V, Yuan Q, Sclaroff S (2005) Simultaneous localization and recognition of dynamic hand gestures. Proceedings of IEEE workshop motion and video computing, vol II. pp 254–260

15. Ke Y, Sukthankar R, Hebert M (2005) Efficient visual event detection using volumetric features. In: Proceedings of 10th IEEE international conference on computer vision, vol 1. pp 166–173
16. Kang H, Lee C, Jung K (2004) Recognition-based gesture spotting in video games. Pattern Recogn Lett 25(15):1701–1714
17. Kahol K, Tripathi P, Panchanathan S (2004) Automated gesture segmentation from dance sequences. In: Proceedings of sixth IEEE international conference on automatic face and gesture recognition, pp 883–888
18. Alon J, Athitsos V, Yuan Q, Sclaroff S (2009) A unified framework for gesture recognition and spatiotemporal gesture segmentation. In: PAMI, September 2009. Video sets available at http://cs-people.bu.edu/athitsos/digits/
19. Lee H, Kim J (1999) An HMM-based threshold model approach for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21(10):961–973
20. Jones M, Rehg J (2002) Statistical color models with application to skin detection. Intl J. Comput Vis 46(1):81–96
21. Alon J (2006) Spatiotemporal gesture segmentation. Ph.D. dissertation, technical report BU-CS-2006-024, Department of Computer Science, Boston University
22. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(27):1–27. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
23. Zhang GP (2000) Neural networks for classification: a survey. IEEE Trans Syst Man Cybern C Cybern 30(4):451–462
24. LeCun Y, Bottou L, Orr GB, Müller KR (1998) Efficient backprop. In: LeCun Y, Bottou L, Orr GB, Muller KR (eds) Neural networks: tricks of the trade, vol 1524. Springer Lecture Notes in Computer Sciences. Springer, Berlin, pp 5–50
25. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE conference on computer vision and pattern recognition, vol I. pp 511–518