J Intell Robot Syst (2014) 76:283–296
DOI 10.1007/s10846-013-9983-7

A Dynamic Gesture and Posture Recognition System

Kyriakos Sgouropoulos · Ekaterini Stergiopoulou · Nikos Papamarkos

Received: 13 June 2013 / Accepted: 27 September 2013 / Published online: 17 October 2014

Abstract This paper presents a real time dynamic hand gesture and posture recognition system based on a neural network and a Hidden Markov Model. For skin color segmentation an adaptive online trained skin color model is used, while the hand posture recognition is accomplished through a likelihood-based classification technique of geometric features. A novel trajectory smoothing technique based on Self Organized Neural Network is introduced to improve HMM classification performance of dynamic gestures. The aim of the proposed system is the creation of a visual dictionary combining hand postures and dynamic gestures. The system has been successfully tested with many people under varying light conditions and different web cameras.

K. Sgouropoulos · E. Stergiopoulou · N. Papamarkos (✉)
Department of Electrical & Computer Engineering, Electric Circuits Analysis Laboratory, Democritus University of Thrace, 67100 Xanthi, Greece
e-mail: papamark@ee.duth.gr
URL: http://www.papamarkos.gr

Keywords Human computer interaction · Hand posture · Dynamic gesture · Skin color detection · Artificial neural network · Hidden Markov Model

1 Introduction

Recent years hand gesture recognition is proven to be an interesting and challenging research field in computer vision. Its main applications are related to Human Computer Interaction (HCI), in order to make the interaction with the computer more natural and user friendly. Moreover, it can be used for the analysis of coverbal gestures in human communication, as well as for the interpretation and learning of sign languages. An overview of the application domains that employ gesture interactions can be found in [1].

There have been suggested several gesture taxonomies according to the author's aspect. Quek et al. [2] proposed the following gesture taxonomy as most appropriate for HCI purposes: (a) Deictic gestures, which point to entities, spatial locations or directions, (b) Manipulative gestures, used to manipulate an entity by the movements of the hand, (c) Semaphoric gestures, used by systems that employ a stylized dictionary of static postures or dynamic hand gestures, (d) Gesticulation, natural form of gesture to aid expression of thoughts, (e) Language gestures, used for sign languages.

The system presented in current work aims to semaphoric gestures recognition.

Hand posture/gesture recognition requires the determination of the hand spatial position, configuration and movement. The methods proposed in literature can be divided into those that use sensing devices attached to the user's hand and into those that use computer vision techniques. Computer vision techniques have gained more attention due to the simplicity of the input devices, which are usually low cost cameras. Most vision based techniques make use of skin color models often combined with depth information acquired from IR-based cameras or stereoscopic cameras [3–5]. The skin color modeling approaches vary and an extensive analysis of them can be found in [6] and [7]. Both surveys deal with skin detection in various color spaces, using different skin modeling techniques, while Kakumanu et al. [6] extend their research in adaptation illumination techniques that use skin-color constancy and dynamic adaptation techniques, like online training using face skin pixels to improve the skin detection performance in dynamically changing illumination and environmental conditions [8, 9]. Gestures can be static, dynamic or both. Concerning static hand posture recognition, model based and appearance based approaches are more frequently encountered [10–12]. An extensive review of hand modeling, analysis and recognition can be found in [13], while Hasan and Mishra [14] focus on hand modeling and posture recognition using geometric features. Classification of dynamic hand gestures recognition involves the use of techniques such as time-compressing templates, dynamic time warping, Hidden Markov Models, and Time Delay Neural Networks, thoroughly described by Mitra in [15].

Van den Bergh and Van Gool [3] propose a combination of Haar wavelets and neural network for a hand gesture recognition system where hand detection is achieved by combining depth information, a pre-trained skin color GMM (Gaussian Mixture Model) and an adaptive histogram-based skin color model which is updated on the fly with color information taken from the face. The proposed system is able to recognize six postures combined with four hand movements. Depth and color information is also used by Elmezain et al. [4] which propose a system capable of recognizing ten dynamic gestures with Hidden Markov Models and Kollorz et al. [5] which present an approach for twelve postures classification. Wang and Wang [16] use the discrete Adaboost learning algorithm with SIFT features for both hand region detection and static posture recognition. Kulkarni and Lokhande [17] present a posture recogntion system for automatic translation of 24 alphabets in American Sign Language using a neural network. Caridakis et al. [18] introduce a dynamic gesture recognition scheme which combines a Self Organized Map neural network and a Hidden Markov model for improving decoding procedure through spatial trajectory classification.

In this paper, a novel hand trajectory smoothing algorithm based on an Artificial Neural Network is introduced as a preprocessing stage for performance improvement of the Hiden Markov Model classifier for gesture recognition. In our approach, in order to increase the number of recognizeable gestures and keep a high detection rate, we combine static postures and dynamic gestures. The proposed technique for real time gesture recognition consists of the following main stages:

- Hand region detection. The hand region detection is based on an adaptive bivariate Gaussian model trained online from pixels of the face, in the color space *YCbCr*, combined with a skin color filtering technique, rather than an offline training skin color model, used in previous works.
- Static hand posture recognition. For posture recognition an improved version of the technique proposed in [19] based on geometric features is used.
- Gesture recognition. For dynamic gesture recognition, a novel trajectory smoothing algorithm based on a ANN is used, followed by a HMM classifier.

The entire system is able to recognize with success 14 postures combined with 10 dynamic gestures resulting to 140 gestures using a low cost web camera. The final system has been successfully tested with different people, web cameras and under varying lighting conditions.

The structure of the paper is as follow: Sections 2, 3 and 4 describe the three main stages of the proposed system, while Section 5 describes the combination of posture and dynamic gestures for lexicon creation. Section 6 presents experimental results of the proposed system. Finally, Section 7 concludes the paper.

2 Hand Region Detection

Accurate hand region detection is important for the successful recognition of the hand posture. For hand region detection, firstly a color filtering procedure is applied in the *YCbCr* color space. This approach is based on a dynamic color modeling of the face skin pixels. Thereafter, connected component analysis is used to extract hand region and finally a gap filling algorithm is applied to eliminate holes existing in the hand image.

2.1 Skin Color Filtering Technique

In the proposed technique, to detect hand regions, we use a skin color model trained by face skin pixels in combination with explicitly defined skin region in *YCbCr* color space. In order to take into account the illumination conditions, the skin color model is adapted after a specific number of frames. We consider that the hand and the face have similar colors in *YCbCr* color space.

2.1.1 Face Detection

For face detection we use a well known technique, which was firstly developed by Viola and Jones [20] and later extended by Lienhart and Maydt [21] and implemented by the Opencv library [22]. This technique has been proven to be robust in identifying face region under various conditions and is extensively used for real time applications [23].

The Viola and Jones detector takes advantage of three techniques, i.e. the "integral image" and the use of cascade and boosted classifiers. The integral image allows very fast feature evaluation which is a necessary condition for real time applications. Specifically, the integral image technique introduced by Viola and Jones permits a rapid computation of simple Haar like rectangular features. The Haar like features used by the algorithm encode the orientation of local contrasts and their spatial relationships, exhibited by a human face. The cascade classifier consists of several simpler classifiers with boosting techniques.

The proposed cascade of boosted classifiers is trained by a large set of sample views of frontal faces, called positive examples and a set of arbitrary images, called negative examples. After the supervised training is completed, the classifier can be applied to a region of interest in an image to check if the region is similar to the positive examples. This classification procedure is independent to the size of the region, since the classifier can be resized instead of resizing the image itself. In order to find an object in the whole image, the search window is moved across the image and every location is checked using the cascade classifier.

The output of the face detector is a rectangle indicating the region of the face, which will be firstly used to create the skin color model and to determine hands blob size and position in a later stage. Figure 1a depicts the result of the face detector algorithm.

2.1.2 Construction of the Skin Color Model

Many skin color models (SCM) have been developed for skin detection during face tracking procedure which can be used also for hand tracking. Dynamic skin color modeling is quite different from static image analysis. First, in principle, the SCM can be less general, i.e. it is trained for one specific person, camera or illumination conditions. Training is possible using the face region which can be discriminated from background by a face classification procedure or manually. This gives a possibility to obtain a skin classification model that is optimal for the given conditions (person, camera, illumination). Since there is no need for a general SCM, with the dynamic skin color modeling approach it is possible to reach higher skin detection rates with low false positives. On the other hand, skin color distribution can vary with time, along with lighting or with camera's white balance change. In this way, the model should be

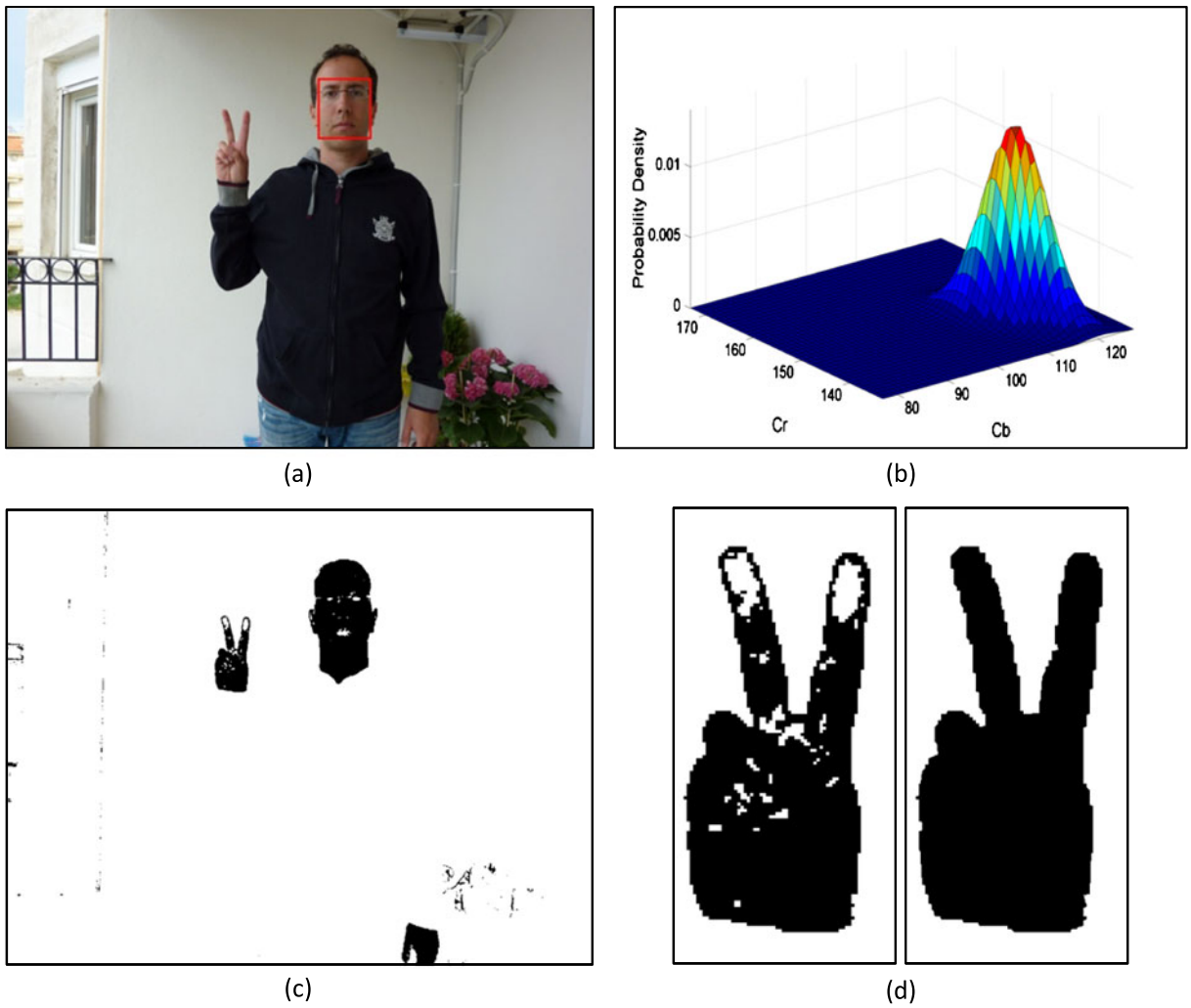

Fig. 1 **a** Face detection; **b** Skin color model distribution; **c** Inverted binary image of the skin color filtering technique; **d** Application of the gap filling filter

able to update itself and to match the changing conditions. Also, model training and classification time becomes extremely important here, because a skin detection system must work at real-time with low computational requirements. For these reasons, many researches use simple parametric skin models that can be easily updated according to the data distribution change, are acceptably fast, and need little storage space. The high rate of false positive results, a common condition of parametric SCM, is not a problem here. The need for specific but not general SCM permits a good classification performance.

In our approach, the SCM is based on a bivariate Gaussian distribution. Yang et al. [24] using goodness-of-fit techniques verified that the skin-color distribution of each individual under a certain lighting condition can be characterized by a multivariate normal distribution. The bivariate Gaussian distribution is defined by the following probability density function:

$$pdf(Cb, Cr) = \frac{1}{2\pi\sigma_{Cb}\sigma_{Cr}\sqrt{1-\rho^2}} \times \exp\left[-\frac{z}{2\left(1-\rho^2\right)}\right] \quad (1)$$

where

$$z = \frac{(Cb - \mu_{Cb})^2}{\sigma_{Cb}^2} - \frac{2\rho\,(Cb - \mu_{Cb})\,(Cr - \mu_{Cr})}{\sigma_{Cb}\sigma_{Cr}} + \frac{(Cr - \mu_{Cr})^2}{\sigma_{Cr}^2} \quad (2)$$

and

$$\rho = correlation\,(Cb\,, Cr) = \frac{V_{Cb\,Cr}}{\sigma_{Cb}\sigma_{Cr}} \quad (3)$$

The pixels of the previously detected face region are used for the calculation of the proper values of μ and σ for each Cb and Cr color components and their covariance matrix. In order to overcome the defect of false face skin pixels, like eyes, hair, glasses etc. we define an explicitly color skin region in Cb-Cr color space. The method is based on the skin color reference map used by Chai and Ngan [25]. They have found that a skin-color region can be identified by the presence of a certain set of chrominance (i.e., Cr and Cb) values narrowly and consistently distributed in the $YCbCr$ color space. The range $\mathrm{R}_{Cb} = [77,127]$ and $R_{Cr} = [133,173]$ is proposed as the most suitable region for a large set of images that have been tested. In our case, the candidate color skin pixels must belong to the above range. That is, a pixel $(i,\ f)$ is considered as possible face skin pixel if $Cr\,(i,\ j) \in R_{Cr} \cap Cb\,(i,\ j) \in R_{Cb}$. Only these pixels are used for the calculation of the SCM distribution. Figure 1b shows the plot of the SCM distribution of skin face pixels in Cb–Cr color space. It can be seen, that for a certain user and fixed lighting condition, color skin range in Cb – Cr color space is quite narrower than the range proposed for general SCM like the one of Chai and Ngan [25]. The above observation explains the fact that dynamic SCMs perform better than general SCMs in terms of false positive skin pixels.

Using, the obtained SCM, the skin probability for all pixels in the image is calculated. A pixel is considered to be skin color if its corresponding probability is greater than a threshold value. Using this SCM, we proceed to skin color segmentation and obtain the skin and non-skin regions. This procedure leads to a binary image, where the skin pixels are depicted in black Fig. 1c. The binary image $BO(i,\ f)$ of the skin color filtering technique can be described by

$$BO(i,j) = \left\{ \begin{matrix} 1,\ if\ \ pdf(Cb_{(i,j)}, Cr_{(i,j)}) \geqslant T \\ 0, \qquad\qquad otherwise \end{matrix} \right\} \quad (4)$$

However, in order to increase the efficiency and speed up the entire procedure of SCM technique an adaptive procedure is introduced. Specifically, after 25 frames the m and σ values of $pdf(Cb,Cr)$ are recalculated, taking into account the previous SCM values. This procedure allows us to tackle effectively changes in brightness conditions.

2.2 Hand Blobs Detection

In the final stage of hand region detection a connected labeling algorithm is applied, where small blobs, compared to the face size, are eliminated and the rest of them are labeled as left or right hand.

The output of the hand region detection technique is a binary image of the hand. A new gap filling filter is applied, in order to eliminate holes existing in the hand image. The filter fills the holes based on the contour detection and flood filling algorithms. The filter firstly detects contours in the binary image using the border following algorithm proposed by Suzuki [26]. The border following algorithm distinguishes between exterior boundaries of objects and interior boundaries, which may be considered as exterior boundaries of holes. The contour detection is followed by flood filling of holes which are enclosed by interior boundaries. The application of the filter results in the removal of holes from the image of the palm, which is essential to the hand posture recognition stage. An experimental result is illustrated in Fig. 1d.

3 Posture Recognition Technique

The next stage deals with the problem of hand posture recognition. This approach is based on defining how many and which fingers appear in the hand image. The entire procedure consists of the following main steps.

3.1 Hand Morphology

To describe the morphology of the hand is necessary to detect some characteristic points of the hand, which are considered as input to the next stage of the finger identification. The description of the hand morphology and its precision is very important and crucial, because it affects the robustness of features used during the recognition process. The hand morphology is satisfactorily described by the palm and its centre, the number of the raised fingers and their tips and roots.

3.1.1 About Palm Region

In order to obtain the palm centre and raised fingers, we find the largest rectangle that can be fitted in the hand. Since the filling holes algorithm has been applied, there are no background pixels in the palm area. The associated dynamic programming algorithm used finds the largest rectangular region containing no background pixels [27]. To do this, the algorithm iterates over each pixel and determines how far to the top, left, and right a border can be extended such that each

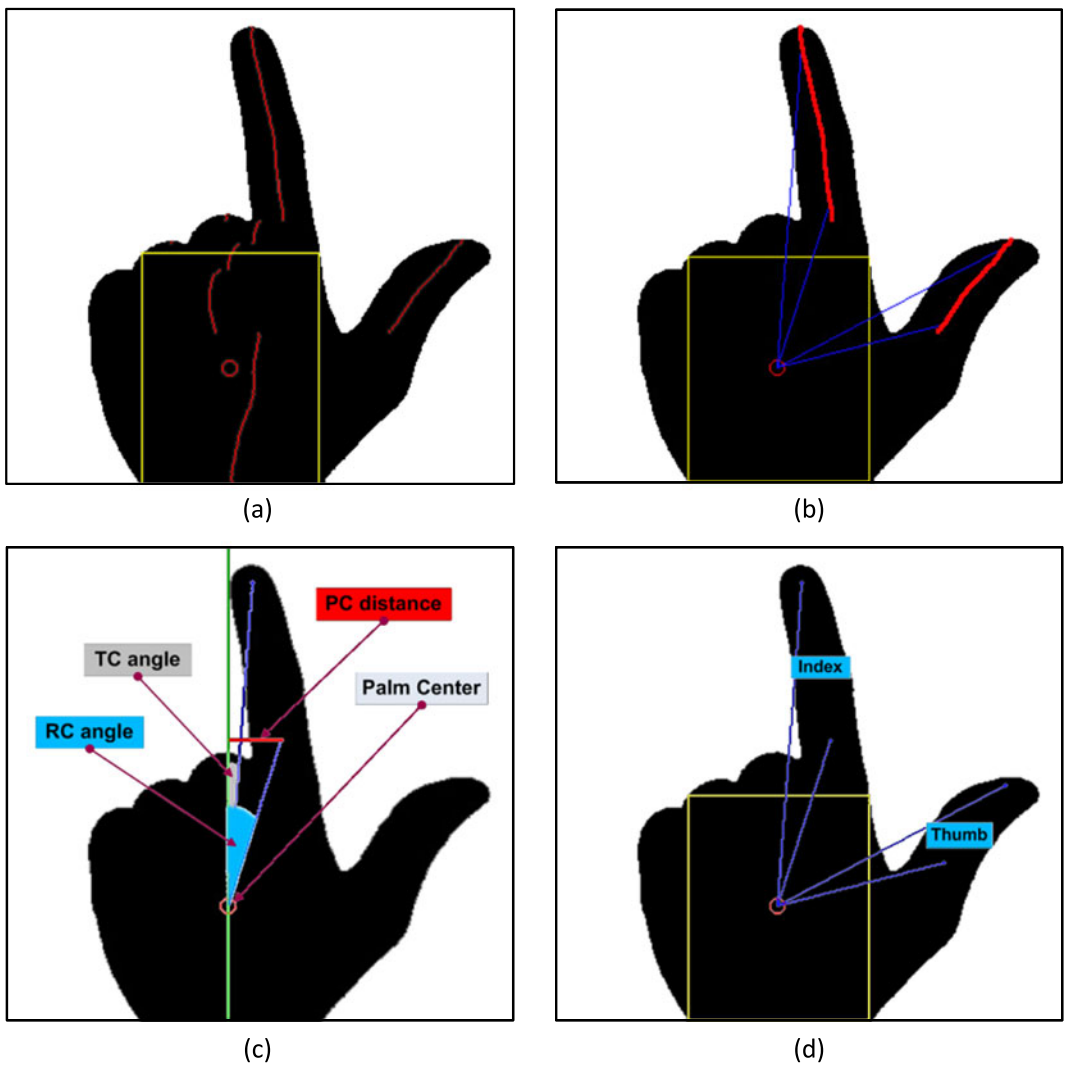

Fig. 2 **a** Local maxima of horizontal distance transform; **b** Root and fingertips points; **c** RC, TC angles and distance from the palm center ; **d** Raised fingers classfication

border marks the first boundary between a hand and background pixel.

The morphology of the hand is such that the rectangular shape detected contains only pixels belonging to the palm (not to fingers). If the hand is closed and no finger is raised, then the rectangle contains almost the entire hand. The palm centre (x_{pc}, y_{pc}) is considered to be equal to centre of the detected rectangular Fig. 2a.

3.1.2 Fingers Definition

The most characteristic points of a raised finger are its tip and its root. In order to spot them, a simple skeletonizing algorithm is performed on the raised fingers.

It is a common practice in a discrete binary image to reduce thick objects to thinner representations called skeletons which satisfactorily characterize the objects. Examples include handwritten or printed characters, circuit diagrams and biological cell structures. In such situations the thickness of the pattern strokes does not contribute to the recognition process [28]. Most skeleton algorithms typically examine the neighborhood of each contour pixel and identify those pixels that can be deleted and those that can be classified as skeleton pixels. Our approach is based on the distance transform algorithm.

Distance transform assigns to each feature pixel of a binary image a value equal to its distance to the nearest no feature pixel. A thinned subset of the original image can be derived from the distance transform by extracting the image that consists of the local maxima of the distance transform. This derived subset is called distance skeleton [29]. We modify the distance transform algorithm by computing the distance to nearest background pixel only in the horizontal axis. Thus, we consider the local maxima only in the horizontal axis.

All the horizontal local maxima pixels form a line for each raised finger. Essentially, this line starts from the base of the finger and ends at the fingertip and consists of the points that have equal horizontal distance from the boundaries of the fingers. Figure 2a depicts the binary image of the hand with the finger lines. Since the algorithm is applied to the whole hand image, a line for the palm is detected, as well some short lines near the roots of the fingers due to morphology of the hand. In the extracted binary image where only thick lines exist, the fingers and the palm have been replaced by lines.

Irregularities of fingers edges, e.g. rapid changes in the thickness of the fingers, may cause finger lines to be discontinued since the skeletonizing algorithm, computes only horizontal local maxima, does not provide any mechanism to ensure their continuality. To ensure that the line of each finger is continuous, morphological dilation operation with a 3-by-3 structuring element is applied on the binary image. We take advantage of the effect of the operator, which is to gradually enlarge the boundaries of regions of foreground pixels, to connect lines which belong to the same finger.

The last step of the hand morphology stage is to determine which lines appearing in the hand image represent fingers and which should be rejected since they represent the palm or irregularities of the hand. Firstly, all line points which are enclosed in the palm rectangle or lying below it, are rejected as they do not belong to fingers. Thereafter, in order to eliminate unnecessary short lines a length size filter is applied which eliminates the lines with length less than 20 percent of palm height. All longer lines which do not belong to fingers have already been rejected as they belong to the palm.

Finally, the output of this stage is the root points $(x_{\text{root}}, y_{\text{root}})$ and the fingertip point $(x_{\text{fingertip}}, y_{\text{fingertip}})$ of each finger, which are extracted from the lines. We define as root point the end point of the line that is closer to the hand centre. The other end point is considered as the fingertip point. Figure 2b depicts the result obtained after the application of the length size filtering procedure and the extraction of roots and fingertips points.

3.2 Posture Recognition

The extracted features describe morphologic and geometric properties of the fingers and can be used to calculate the following three features:

- RC Angle. This is considered as the first feature. Actually this feature corresponds to an angle formed by the vertical axis and the line that joints the root point and the palm centre.

This angle provides the most discrete values for each finger and thus is valuable for the recognition.

$$R\hat{C} = 90 - \tan^{-1}\left(\frac{y_{\text{root}} - y_{\text{pc}}}{x_{\text{root}} - x_{\text{pc}}}\right) \quad (5)$$

- TC Angle. The second feature is an angle formed by the vertical axis and the line that joints the fingertip point and the palm centre. It is used directly for the finger identification process.

$$T\hat{C} = 90 - \tan^{-1}\left(\frac{y_{\text{fingertip}} - y_{\text{pc}}}{x_{\text{fingertip}} - x_{\text{pc}}}\right) \quad (6)$$

- Distance from the palm centre. The third feature is the vertical distance of the finger's root point from the vertical line passing through the palm centre. The feature is invariant to the size of the hand, because its value is divided by the length of the palm. The length of the palm is defined as the width of the palm rectangle.

RC angle, TC angle and distance from the palm centre for the index finger are shown in Fig. 2c. Using the above three features and the classification process proposed by Stergiopoulou and Papamarkos [19] we can classify the raised fingers into five classes (thumb, index, middle, ring, little) Fig. 2d.

4 Gesture Recognition

Hand posture recognition stage is followed by the hand gesture recognition. The proposed system is able to recognize gestures carried out using a single hand. The recognition is performed through the motion trajectory of the hand using HMM. The main stages of the gesture recognition are the hand tracking, trajectory smoothing, feature extraction and gesture classification.

4.1 Hand Tracking

Using the technique described in Section 2, the hand blob and its location can be detected. The hand position is determined by the coordinates of the centroid of the hand blob. Thus, having determined the hand centroid in every frame, we are able to track the hand in a sequence of frames. The motion path is created by connecting the sequential positions of the hand centroid. So the gesture's trajectory, lasting T frames, is considered the curve that pass through the hand centroid points $C_i(x_i, y_i), i = 0,1,\ldots,T$.

4.2 Trajectory Smoothing

The success of gesture recognition using HMM assumes a rather smooth trajectory. We have found that the trajectory can be satisfactory smoothed by using the SGONG neural network [30]. The SGONG is a self growing and self organized neural gas. It achieves clustering of the input data, so as the distance of the data items within the same class (intra-cluster variance) is small and the distance of the data items stemming from different classes (inter-cluster variance) is large. It is an innovative neural network that combines the advantages both of the Kohonen Self-Organized Feature Map (SOFM) [31] and the Growing Neural Gas (GNG) [32, 33], neural classifiers according to which, the learning rate and the radius of the neighborhood domain of neurons is monotonically decreased during the training procedure.

A main advantage of the SGONG classifier as it can adaptively determine the final number of neurons. This characteristic permits SGONG to capture efficiently the feature space. The SGONG classifier has already been used for the identification of geometric characteristics [19] and color reduction [30]

Trajectory smoothing is achieved by feeding the SGONG with the coordinates of a set of subsampled points of the motion path that is defined by the T hand centroid points of the gesture. The maximum number of output neurons depends on the most complex gesture that has to be recognized, since the network should approximate the path without discarding significant parts of the motion path. During the training phase the neural tries to fit the trajectory using an appropriate number of neurons. When the network has been trained, each hand centroid point is replaced by the position of the nearest output neuron using the Euclidean distance. Figure 3 depicts the results of the application of the SGONG on a number

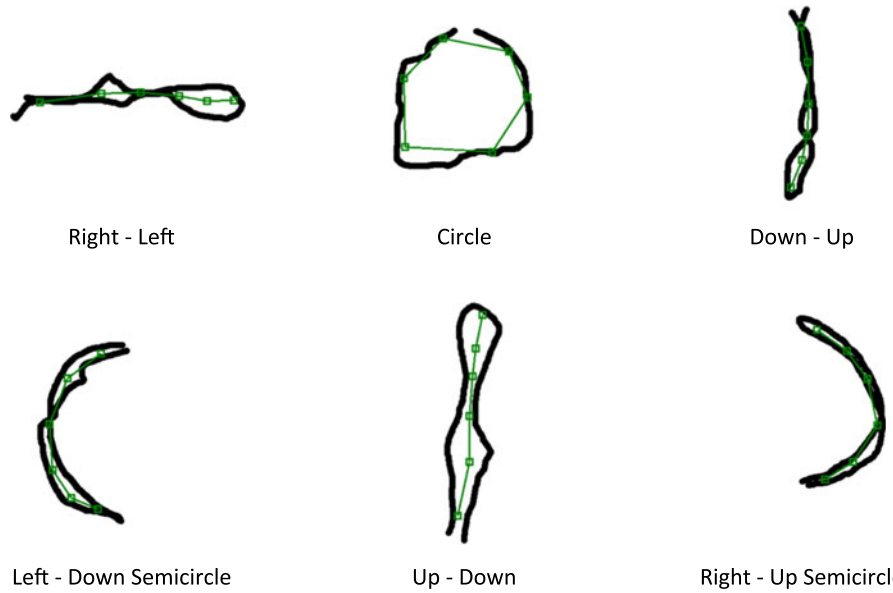

Fig. 3 Application of the SGONG Smoothing algorithm on noisy trajectories. Positions of the SGONG neurons are depicted by *green cirlces* and neurons' connections by *green lines*

of noisy motion paths. As we can observe from these results the approximation of the trajectory using SGONG leads to an adequate smoothing of the initial trajectory. Now the new motion path is defined by the new positions given by the output neurons Fig. 4.

4.3 Gesture Classification

An HMM [34] consists of a finite number of hidden states with known transition probabilities between them (Markov chain). The states are hidden because they cannot be directly seen, except

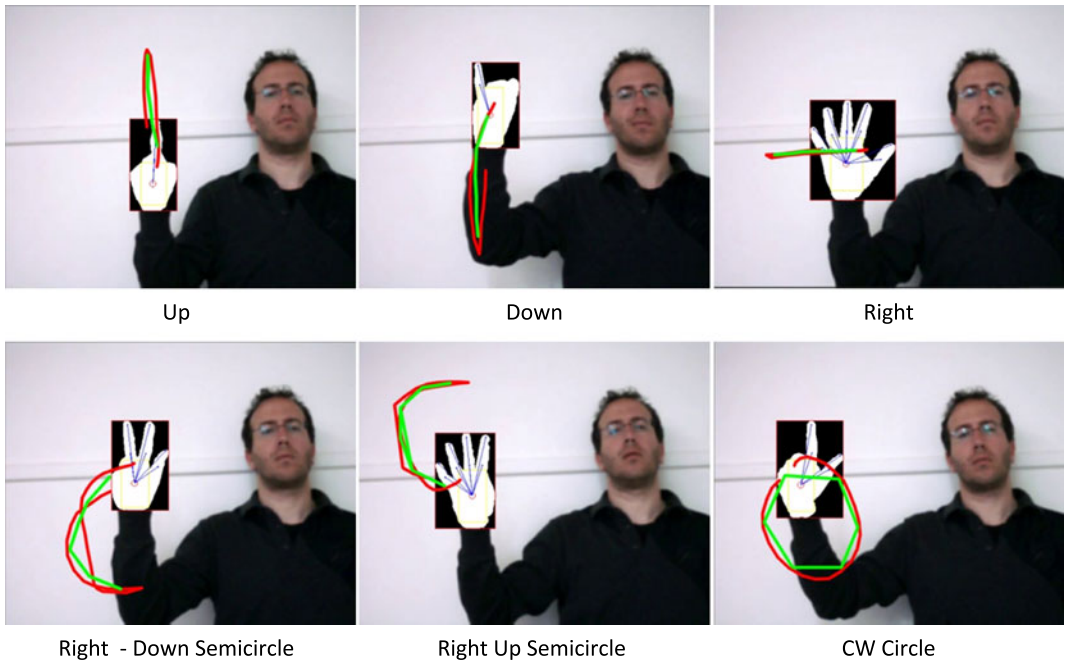

Fig. 4 Detected tranjectory (*red line*)—SGONG smoothed trajectory (*green line*)

through an observation (output) generated by each state according to the associated probability distribution.

The HMMs are classified according their topology. When, any state can be reached from any other state, then it's a fully connected structure, known as an *ergodic* model. Another topology is the Left—Right (LR) model, where a transition is possible only to the same or following states. The LR models have been proven to perform better than other topologies for gesture recognition because they are restricted, simple and therefore make it easier for the training algorithm to extract the most information out of the data [35]. A three-state LR HMM is depicted in Fig. 5.

A HMM can be expressed by $\lambda = (A, B, \Pi)$ where:

- $A = \{a_{i.j}\}$ is the $N \times N$ state transition probabilities matrix, N is the number of the model's states $S = \{s_1, s_2, \cdots, s_N\}$
- $B = \{b_{j,k}\}$ is the $N \times M$ observation probability matrix, M is the number of discrete observation symbols $V = \{v_1, v_2, \cdots, v_M\}$.
- $\Pi = \{\pi_i\}$ is a N-dimensional vector of the initial probability distribution for the model's states.

An observation sequence is defined as $O = O_1, O_2, \cdots O_T$, where T is the total number of observations in the sequence and each observation O_t is one of the symbols of V.

There are three problems associated with HMM and three solutions that have been proposed [34]:

1. *evaluation*: given the model $\lambda = (A, B, \Pi)$ and an observation sequence O, find the conditional probability $P(O|\lambda)$. It can be efficiently solved using the Forward algorithm.
2. *decoding*: given the model $\lambda = (A, B, \Pi)$ and an observation sequence O, find an optimal state sequence for the underlying Markov process, i.e. recover the state sequence. It can also be solved using the Viterbi algorithm.
3. *training*: given an observation sequence O and the dimensions N and M, find the parameters of the model $\lambda = (A, B, \Pi)$, that maximize the probability of O. The problem cannot be exactly solved, but the Baum-Welch algorithm can derive a local maximum likelihood.

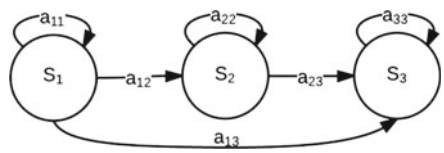

Fig. 5 3-state *left–right* Hidden Markov Model

Concerning the gesture recognition problem, the third and the first solution are used. Firstly the number of states N of the HMM and the number of the distinguish symbols M are determined according to the complexity of the gestures that have to be recognized. Then, the solution to the third problem is applied in order to train a set of HMMs $\lambda_1, \lambda_2, \cdots, \lambda_p,$, each one corresponding to the $G_1, G_2, \cdots, G_p$ discrete known gestures that need to be recognized by the system. During the training, each HMM is fed with the train sequences of data belonging to the corresponding gesture. Finally, given an unknown captured gesture, the solution to the first problem is used to score the gesture against λ_o to λ_p to determine whether it is one of the known gestures or neither of them.

Before using HMM for gesture recognition a Vector Quantization stage is necessary. It is used to provide single discrete representation of the continuous features extracted from each gesture. There exist several features that can describe a gesture such as location, distance from the head, orientation and velocity. In our approach the orientation is used as the main feature, since it can be easily extracted and has been used with good results in previous researches [4, 35–37]. The orientation is determined between two consecutive points from hand gesture path by the following equation:

$$\theta_t = \tan^{-1}\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right) \quad t = 1, 2, \ldots, T-1 \qquad (7)$$

where T represents the length of the gesture path. The orientation θ_t is quantized into M classes by dividing it by $Q = \frac{360}{M}$ to form the codebook consisting of M codewords. The Vector Quantization stage introduces a quantization error since we represent an entire arc of angles by a single

Table 1 Set of postures recognized by the proposed system

Posture	#Fingers	Combination	Little	Ring	Middle	Index	Thumb
1	1	0	X	–	–	–	–
2	1	3	–		–	X	–
3	1	4	–	–	–	–	X
4	2	01	X	X	–	–	–
5	2	03	X	–	–	X	–
6	2	04	X	–	–	–	X
7	2	23	–	–	X	X	–
8	2	34	–	–	–	X	X
9	3	034	X	–	–	X	X
10	3	123	–	X	X	X	–
11	3	234	–	–	X	X	X
12	4	0123	X	X	X	X	–
13	4	1234	–	X	X	X	X
14	5	12345	X	X	X	X	X

angle. It can be seen that the greater the size of the codebook the smaller the quantization error. On the other hand, a large codebook leads to problems implementing HMMs with a large number of parameters [34]. In the most cases, the size M of the codebook is experimentally determined.

5 Lexicon Creation

In the proposed framework, static hand postures are combined with hand gestures to formulate coded gestures (words). Hand postures are followed by hand movements leading to distinct words. In detail, the user, before performing a gesture, keeps his hand still and forms a specific posture for at least two seconds (30 frames) using a combination of raised fingers as described in Section 3. After this period, a hand movement is being performed by the user. A single gesture is completed when the hand stays still again for forming another posture.

For as long as the user's hand stays still the system identifies the formed posture. During the

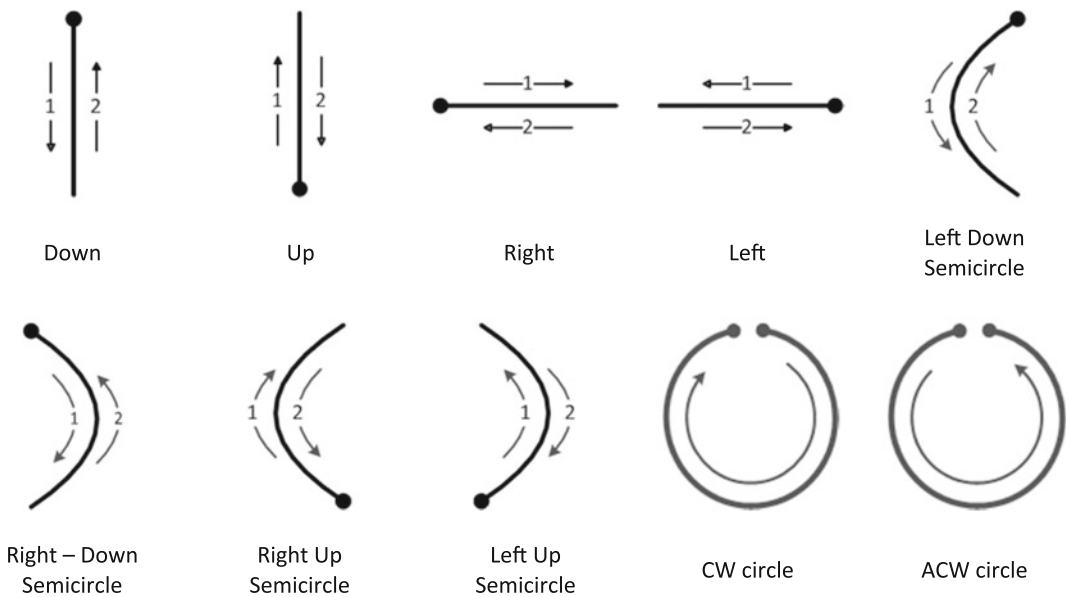

Fig. 6 The gesture dataset

Table 2 Evaluation of posture recognition technique

Posture	1	2	3	4	5	6	7
Rec. rate	88.33 %	96.67 %	83.33 %	88.33 %	96.67 %	91.67 %	98.33 %
Posture	8	9	10	11	12	13	14
Rec. rate	96.67 %	95.00 %	98.33 %	98.33 %	96.67 %	98.33 %	100.00 %

two seconds when the user's hand remains still the identified postures are not the same, because during this time interval there is movement of user's fingers to form the final posture. The dominant posture between two consecutive gestures is determined in the following way. Firstly, for the posture recognized in each frame, the total number of raised fingers and their classes is determined (e.g. number of fingers 3—thumb, index, middle). After all postures have been recognized, and before starting the next gesture, the total number n of fingers that have been appeared more often is calculated. Then, having sorting the classes of the fingers by the number of times that have appeared, the dominant posture is obtained by the top n classes. The described technique has been proven to perform better than simply choose the posture that appears the most.

The proposed system is able to recognize 14 distinct postures with high recognition rate. The 14 combinations of raised finger are shown in Table 1. Each static hand posture is followed by a gesture. The proposed system is trained to recognize 10 distinct gestures that lead to 140 different words, if combined with the 14 distinct postures. The 10 distinct gestures are shown in Fig. 6.

6 Experimental Results

In order to determine the performance of the proposed system an evaluation was conducted. In the video database that has been collected for the evaluation three different users perform twice 14 sets for each of the 10 gestures each one combined with different hand posture, that is a total number of $3 \times 2 \times 14 \times 10 = 840$ videos. The proposed system was implemented using C# in Intel Core 2 Duo PC with 3GB RAM and a web camera capturing 15 frames/ sec with 640×480 video resolution.

6.1 Experiment 1: Posture Recognition

The evaluation was firstly conducted separately for hand postures and gesture recognition since the latter involves training. The test data for hand posture recognition comprised of all 840 postures appeared in the video database. For the hand postures the average recognition achieved was 94.76 %. The recognition ratio is the number of correctly identified static hand postures over the number of test hand postures. The high recognition rate was succeeded, mainly, due to the real time feedback of the posture recognized which allows the user to adjust the angle of his hand. Recognition rate for each posture is shown in Table 2. Comparative results to other methods proposed in literature according to their evalutation results is shown Table 3.

6.2 Experiment 2: Gesture Recognition

For gesture recognition experiments were performed to determine firstly the appropriate number of stages N of the HMMs and then the size of the codebook M, using stratified 10-fold cross

Table 3 Comparison of posture recognition rate to other methods presented in relative work

Method	Number of recognized postures	Average recognition rate (%)
[16]	3	95.6
[6]	26	92.78
[5]	14	93.14
[3]	6	99.54
Proposed	14	94.76

Table 4 Dynamic gesture recognition rate against different codebook sizes

Size of codebook M	SGONG smoothing (3 stages) (%)	MF smoothing (3 stages) (%)
6	63.02	67.02
8	84.61	81.08
12	73.10	72.39

Table 5 Dynamic gesture recognition rate against different number of stages of the Hiddem Markov Model

Number of stages N	SGONG smoothing (%)	MF smoothing (%)
3	84.61	81.08
5	90.83	85.14
8	92.66	79.39
10	91.74	78.38
12	91.74	76.69

validation strategy, recommended by Kohavi [38], as it provides less biased estimation of the accuracy. The proposed technique of SGONG for trajectory smoothing was tested against the angle median filtering smoothing procedure used by Carydakis et al. [18]. The results of these experiments are presented in Tables 4 and 5.

7 Conclusions

This paper presents a new technique for dynamic gesture and posture recognition. The proposed system uses a skin color detection algorithm, a HMM for dynamic gesture recognition, and a geometric features based likelihood classification technique for posture recognition. Also, in order to improve the performance of the HMM classifier the SGONG is used to fit the hand motion trajectories. The proposed technique for skin detection is proven to be more adaptive to lighting variations and has less false positive skin pixels in a complex background. The importance of the proposed technique is that it can be used in real-time with a low-cost web camera. The experimental results indicate the performance of the entire system can be compared with similar approaches that use high quality 3D cameras, such as ToF cameras which provide depth information.

References

1. Murthy, G., Jadon, R.: A review of vision based hand gestures recognition. Int. J. Inf. Technol. Knowl. Manag. **2**(2), 405–410 (2009)
2. Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-F., Kirbas, C., McCullough, K.E., Ansari, R.: Multimodal human discourse: gesture and speech. ACM Trans. Comput-Hum. Interact. (TOCHI). **9**(3), 171–193 (2002)
3. Van den Bergh, M., Van Gool, L.: Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In: 2011 IEEE Workshop on Applications of Computer Vision (WACV) (2011)
4. Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B.: A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In: 19th International Conference on Pattern Recognition, ICPR (2008)
5. Kollorz, E., Penne, J., Hornegger, J., Barke, A.: Gesture recognition with a time-of-flight camera. Int. J. Intell. Syst. Technol. Appl. **5**(3), 334–343 (2008)
6. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. Pattern Recognit. **40**(3), 1106–1122 (2007)
7. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: Proc. Graphicon (2003)
8. Doulamis, N., Doulamis, A., Kosmopoulos, D.: Content-based decomposition of gesture videos. In: IEEE Workshop on Signal Processing Systems Design and Implementation (2005)
9. Araki, R., Gohshi, S., Ikenaga, T.: Real-time both hands tracking using CAMshift with motion mask and probability reduction by motion prediction. In: Signal & Information Processing Association Annual Summit and Conference, (APSIPA ASC) 2012. Asia-Pacific (2012)
10. Oikonomidis, I., Kyriazis, N., Argyros, A.: Efficient model-based 3d tracking of hand articulations using kinect. In: British Machine Vision Conference (2011)
11. Athitsos, V., Wang, H., Stefan, A.: A database-based framework for gesture recognition. Pers. Ubiquit. Comput. **14**(6), 511–526 (2010)
12. Malassiotis, S., Strintzis, M.: Real-time hand posture recognition using range data. Image Vis. Comput. **26**(7), 1027–1037 (2008)
13. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: a review. Comp. Vision Image Underst. **108**(1), 52–73 (2007)
14. Hasan, M.M., Mishra, P.K.: Hand gesture modeling and recognition using geometric features: a review. Can. J. Image Process. Comput. Vis. **3**(1), 12–26 (2012)
15. Mitra, S., Acharya, T.: Gesture recognition: a survey. Syst. Man Cybern. C Appl. Rev. IEEE Trans. **37**(3), 311–324 (2007)
16. Wang, C.-C., Wang, K.-C.: Hand posture recognition using adaboost with SIFT for human robot interaction. In: Recent Progress in Robotics: Viable Robotic Service to Human, pp. 317–329. Springer (2008)
17. Kulkarni, V.S., Lokhande, S.: Appearance based recognition of american sign language using gesture segmentation. Int. J. Comput. Sci. Eng. **2**(3), 560–565 (2010)
18. Caridakis, G., Karpouzis, K., Drosopoulos, A., Kollias, S.: SOMM: self organizing Markov map for gesture recognition. Pattern Recog. Lett. **31**(1), 52–59 (2010)

19. Stergiopoulou, E., Papamarkos, N.: Hand gesture recognition using a neural network shape fitting technique. Eng. Appl. Artif. Intell. **22**(8), 1141–1158 (2009)
20. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE (2001)
21. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: IEEE (2002)
22. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media (2008)
23. Zhang, C., Zhang, Z.: A survey of recent advances in face detection. Microsoft Research (2010)
24. Yang, J., Lu, W., Waibel, A.: Skin-color modeling and adaptation. In: Computer Vision—ACCV'98, pp. 687–694 (1997)
25. Chai, D., Ngan, K.: Face segmentation using skin-color map in videophone applications. IEEE Trans. Circ. Syst. Video Technol. **9**, 551–564 (1999)
26. Suzuki, S., and others: Topological structural analysis of digitized binary images by border following. Comput. Vis. Graph. Image Process. **30**(1), 32–46 (1985)
27. Freeman, E., Brewster, S.: Messy tabletops: clearing up the occlusion problem. In: CHI'13 Extended Abstracts on Human Factors in Computing Systems (2013)
28. Jain, A.: Fundamentals of Digital Image Processing. Prentice-Hall, Inc. (1989)
29. Ritter, G., Wilson, J.: Handbook of Computer Vision Algorithms in Image Algebra. CRC (2001)
30. Atsalakis, A., Papamarkos, N.: Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas. Eng. Appl. Artif. Intell. **19**(7), 769–786 (2006)
31. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
32. Fritzke, B., others: A growing neural gas network learns topologies. Adv. Neural Inf. Process. Syst. **7**, 625–632 (1995)
33. Fritzke, B.: Growing cell structures—a self-organizing network for unsupervised and supervised learning. Neural Netw. **7**(9), 1441–1460 (1994)
34. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**, 257–286 (1989)
35. Liu, N., Lovell, B.C., Kootsookos, P.J., Davis, R.I.: Model structure selection & training algorithms for an HMM gesture recognition system. In: IEEE (2004)
36. Wilson, A.D., Bobick, A.F.: Parametric hidden markov models for gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **21**, 884–900 (1999)
37. Abdul, Y.F., Wong, F.: Hidden Markov Model-based gesture recognition with overlapping hand-head/hand-hand estimated using kalman filter. In: IEEE (2012)
38. Kohavi, R., others: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence (1995)