Engineering Applications of Artificial Intelligence 22 (2009) 1141–1158

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Hand gesture recognition using a neural network shape fitting technique

E. Stergiopoulou, N. Papamarkos [*]

Image Processing and Multimedia Laboratory, Department of Electrical & Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

ARTICLE INFO

Article history:
Received 8 September 2006
Received in revised form
19 January 2009
Accepted 16 March 2009
Available online 15 May 2009

Keywords:
Human–machine interaction
Hand gesture
Skin color detection
YCbCr color space
Neural network

ABSTRACT

A new method for hand gesture recognition that is based on a hand gesture fitting procedure via a new Self-Growing and Self-Organized Neural Gas (SGONG) network is proposed. Initially, the region of the hand is detected by applying a color segmentation technique based on a skin color filtering procedure in the *YCbCr* color space. Then, the SGONG network is applied on the hand area so as to approach its shape. Based on the output grid of neurons produced by the neural network, palm morphologic characteristics are extracted. These characteristics, in accordance with powerful finger features, allow the identification of the raised fingers. Finally, the hand gesture recognition is accomplished through a likelihood-based classification technique. The proposed system has been extensively tested with success.

1. Introduction

Hand gesture recognition is a promising research field in computer vision. Its most appealing application is the development of more effective and friendly interfaces for human–machine interaction, since gestures are a natural and powerful way of communication. Moreover, it can be used for teleconferencing because it does not require any special hardware. Last but not least, it can be applied to the interpretation and learning of sign languages.

Hand gesture recognition is a complex problem that has been dealt with in many different ways. Kjeldssen and Kender (1996) suggest an algorithm of skin color segmentation in the HSV color space and use a backpropagation neural network to recognize gestures from the segmented hand images. Huang and Huang (1998) propose a system consisting of three modules: (i) model-based hand tracking that uses the Hausdorff (Huttenlocher et al., 1992) distance measure to track shape-variant hand motion, (ii) feature extraction by applying the scale and rotation invariant Fourier descriptors and (iii) recognition by using a 3D modified Hopfield neural network. Hongo et al. (2000) use a skin color segmentation technique in order to segment the region of interest and then recognize the gestures by extracting directional features and using linear discriminant analysis. Manresa et al. (2000) propose a method of three main steps: (i) hand segmentation based on skin color information, (ii) tracking of the position and the orientation of the hand by using a pixel-based tracking for the temporal update of the hand state and (iii) estimation of the hand state in order to extract several hand features to define a deterministic process of gesture recognition. Huang and Jeng (2001) suggest a model-based recognition system that also consists of three stages: (i) feature extraction based on spatial (edge) and temporal (motion) information, (ii) training that uses the Principal Component Analysis, the Hidden Markov Model (HMM) and a modified Hausdorff distance and (iii) recognition by applying the Viterbi algorithm. Herpers et al. (2001) use a hand segmentation algorithm that detects connected skin–tone blobs in the region of interest. A medial axis transform is applied, and finally, an analysis of the resulting image skeleton allows the gesture recognition. Yoon et al. (2001) propose a system consisting of three different modules: (i) hand localization, (ii) hand tracking and (iii) gesture spotting. The hand location module detects hand candidate regions on the basis of skin color and motion. The hand tracking algorithm finds the centroids of the moving hand regions, connects them, and produces a hand trajectory. The gesture spotting algorithm divides the trajectory into real and meaningless segments. This approach uses location, angle and velocity feature codes, and employs a *k*-means clustering algorithm for the HMM codebook. Triesch and Von der Malsburg (2001) propose a computer vision system that is based on Elastic Graph Matching, which is extended in order to allow combinations of different feature types at the graph nodes. Chen et al. (2003) introduce a hand gesture recognition system to recognize continuous gesture before stationary background. The system consists of four modules: a real-time hand tracking and extraction, feature extraction, HMM training, and gesture recognition. First, they apply a real-time hand tracking and extraction

[*] Corresponding author. Tel.: +30 25410 79585; fax: +30 25410 79569.
E-mail address: papamark@ee.duth.gr (N. Papamarkos).
URL: http://www.papamarkos.gr/ (N. Papamarkos).

0952-1976/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2009.03.008

algorithm to trace the moving hand and extract the hand region, and then they use the Fourier descriptors to characterize spatial features and the motion analysis to characterize the temporal features. They combine the spatial and temporal features of the input image sequence as the feature vector. After having extracted the feature vectors, they apply HMMs to recognize the input gesture. The gesture to be recognized is separately scored against different HMMs. The model with the highest score indicates the corresponding gesture. Xiaoming and Ming (2003) use an RCE neural network-based color segmentation algorithm for hand segmentation, extract edge points of fingers as points of interest and match them based on the topological features of the hand, such as the center of the palm. Tan and Davis (2004) track the face and hand regions using color-based segmentation and Kalman filtering. Next, different classes of natural hand gesture are recognized from the hand trajectories by identifying gesture holds, position/velocity changes, and repetitive movements. According to the method proposed by Doulamis et al. (2005), the gesture segmentation is performed based on skin color information, the segmented regions are represented using the Zernike moments and finally an adaptive hierarchical content decomposition algorithm is applied. Wachs et al. (2005) identify static hand gesture poses by using Haar-like features to represent the shape of the hand. These features are used as input to a fuzzy *c*-means clustering algorithm for pose classification. A probabilistic neighborhood search algorithm is employed to automatically select a small number of Haar features, and to tune the fuzzy *c*-means classification algorithm. Licsar and Sziranyi (2005) use a background subtraction method in order to accomplish hand segmentation and classify the static hand gestures based on the Fourier descriptors. The recognition method consists of a supervised and an unsupervised training procedure. Finally, a new technique for shape-based hand recognition is proposed by Yoruk et al. (2006).

In the proposed method, hand gesture recognition is divided into four main stages: the detection of the hand's region, the approximation of its shape, the extraction of its features, and finally its identification. The detection of the hand's region is achieved by using a color segmentation technique based on a skin color distribution map in the *YCbCr* space (Chai and Ngan, 1998, 1999). The technique is reliable, since it is relatively immune to changing lighting conditions and provides good coverage of the human skin color. It is very fast and does not require post-processing of the hand image. Once the hand is detected, a new Self-Growing and Self-Organized Neural Gas (SGONG) (Atsalakis and Papamarkos, 2005a, b, 2006; Atsalakis et al., 2005) neural network is used in order to approximate its shape. The SGONG is an innovative neural network that grows according to the hand's morphology in a very robust way. As it is shown in Fig. 1(a), the SGONG starts with only two neurons and grows up until its convergence (Fig. 1(b)). In Fig. 1(c) it is obvious that the grid of the output neurons takes the shape of the hand. Also, an effective algorithm is developed in order to locate the gesture's raised fingers, which is a necessary step for the recognition process. In the final stage, suitable features are extracted that identify, regardless to the hand's slope, the raised fingers. Finally, the completion of the gesture's recognition process is achieved by using a likelihood-based classification method.

The proposed gesture recognition system has been trained to identify 31 hand gestures that derive from the combination of raised and not raised fingers. This set of gestures can be used for human–computer communication without the interference of any special hardware. It has been tested by using a large number of input images and the achieved recognition rate is very promising. A short version of the proposed technique is accepted for presentation in ICIP2006 (Stergiopoulou and Papamarkos, 2006).

2. Description of the method

The purpose of the proposed gesture recognition method is to recognize a set of 31 hand gestures. The principal assumption is that the input images include exactly one hand. Furthermore, the gestures are made with the right hand, the arm is roughly vertical, the palm is facing the camera and the fingers are either raised or not. Finally, the image background is plain and uniform.

The entire method consists of the following four main stages:

Stage 1: Hand region detection.

Stage 2: Approximation of the hand's morphology.

Stage 3: Finger identification.

Stage 4: Recognition process.

Analysis of these stages follows.

2.1. Hand region detection

The first step of a hand recognition process is the detection of the hand region. In the proposed method, this is achieved through color segmentation, i.e. classification of the pixels of the input image into skin color and non-skin color clusters. The technique is based on color information, because color is a highly robust feature. First of all, it is invariant to rotation and scaling as well as to morphologic variations of the hand. Secondly and importantly it allows a simple and fast processing of the input image. On the other hand, skin color varies quite dramatically. It is vulnerable to changing lighting conditions and it differs among people and especially among people from different ethnic groups. The perceived variance, however, is really a variance in luminance due

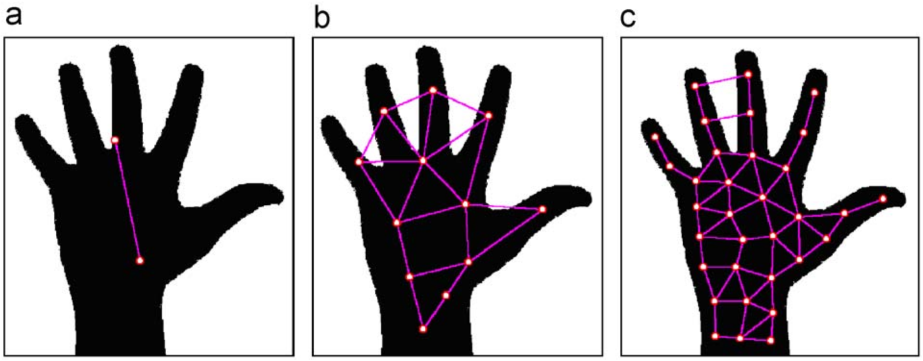


Fig. 1. Growth of the SGONG network: (a) starting point, (b) a growing stage and (c) the final output grid of neurons.

to the fairness or the darkness of the skin. Moreover, researchers claim that skin chromaticity is roughly invariant among different races (O'Mara, 2002; Albiol et al., 2001). So regarding skin color, luminance introduces many problems, whereas chromaticity includes useful information. Therefore, skin color detection is possible and successful by using proper color spaces that separate luminance from chromaticity components.

2.1.1. *YCbCr color space*

The proposed hand region detection technique is applied in the *YCbCr* color space. *YCbCr* was created as part of ITU-R BT.601 during the development of a world-wide digital component video standard. It is a television transmission color space and sometimes is known as a transmission primary. It is device dependent and also quite unintuitive. *YCbCr* is useful in compression applications and most importantly it separates *RGB* into luminance and chrominance information. In particular, *Y* is the luminance component and *Cb*, *Cr* are the chrominance components. *RGB* values can be transformed to *YCbCr* color space using the following equation:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (1)$$

Given that the input *RGB* values are within range [0,1] the output values of the transformation will be in the ranges [16, 235] for *Y* and [16, 240] for *Cb* and *Cr*. Fig. 2 shows the histograms of the *Y*, *Cb* and *Cr* components of three different skin color hands: (a) a white hand poorly illuminated, (b) a white hand well illuminated and (c) a black hand well illuminated. As it was marked previously, the *Y* component varies greatly whereas the *Cb* and *Cr* components are approximately the same for the three input images. Consequently, the *YCbCr* color space is indeed a proper space for skin color detection.

2.1.2. *Skin color detection technique*

The classification of the pixels of the input image into skin color and non-skin color clusters is accomplished by using a

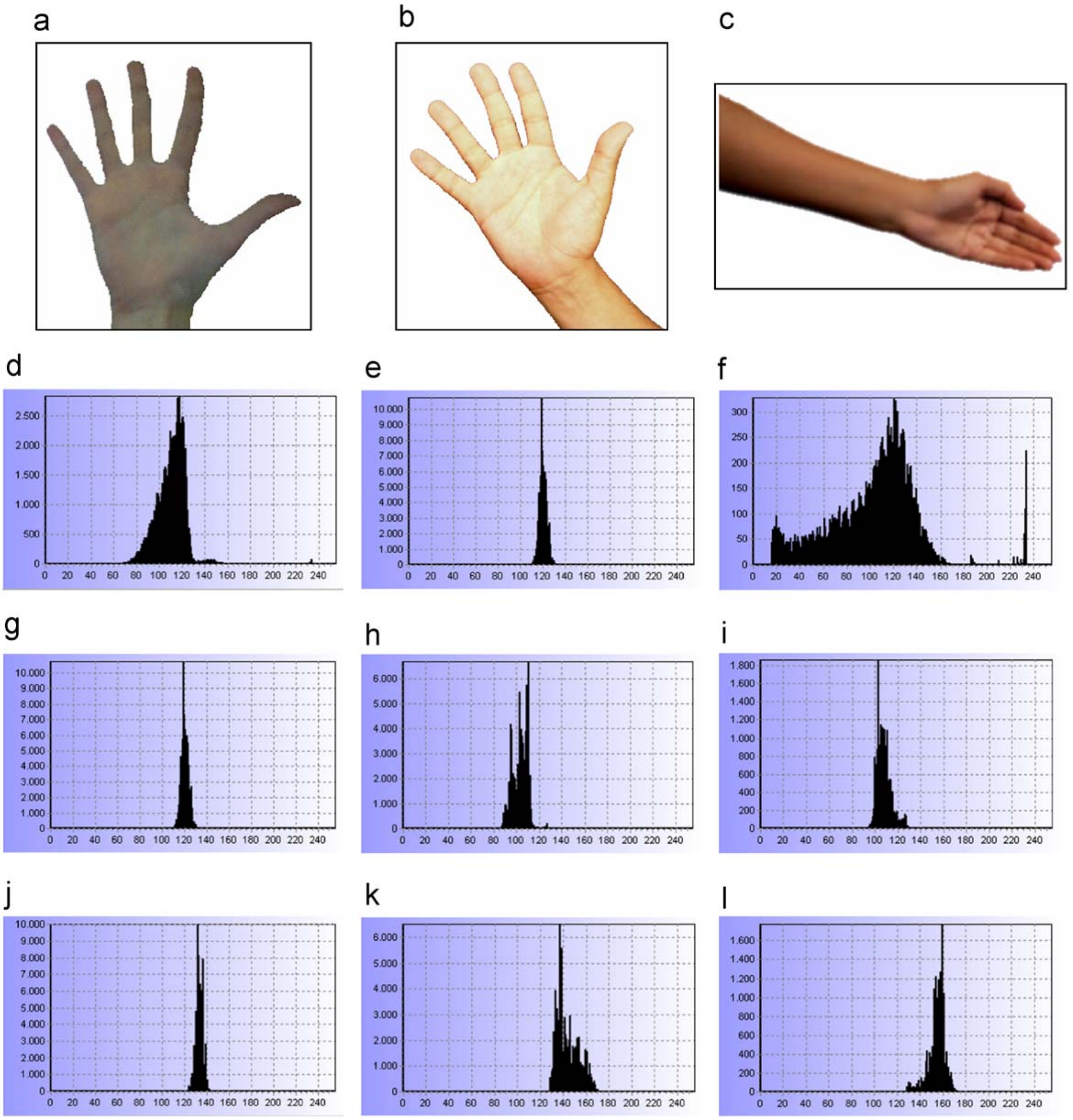


Fig. 2. (a) Input image of a white hand poorly illuminated, (b) input image of a white hand well illuminated, (c) input image of a black hand well illuminated, (d)–(f) *Y* component histograms, (g)–(i) *Cb* component histograms and (j)–(l) *Cr* component histograms.

thresholding technique that exploits the information of a skin color distribution map in the *YCbCr* color space.

In this method, which is a modification of the Chai and Ngan method (Chai and Ngan, 1998, 1999), a map of the chrominance components of skin color was created by using a training set of 50 images. It was found that *Cb* and *Cr* values are narrowly and consistently distributed. Particularly, the ranges of *Cb* and *Cr* values are, as shown in Fig. 3, $R_{Cb}=[80,105]$ and $R_{Cr}=[130,165]$, respectively. These ranges were selected very strictly, in order to minimize the noise effect and maximize the possibility that the colors correspond to skin.

The steps of the skin color detection technique are the following: Let $Cb^{(i,j)}$ and $Cr^{(i,j)}$ be the chrominance components of the (i,j) pixel.

Step 1: Comparison of the $Cb^{(i,j)}$ and $Cr^{(i,j)}$ values with the R_{Cb} and R_{Cr} ranges. If $Cb^{(i,j)} \in R_{Cb}$ and $Cr^{(i,j)} \in R_{Cr}$, then the pixel belongs to the hand region.

Step 2: Calculation of the Euclidean distances between the $Cb^{(i,j)}$, $Cr^{(i,j)}$ values and the limits of the R_{Cb} and R_{Cr} ranges, for every pixel:

$$\begin{aligned}
D_1 &= \|(Cb^{(i,j)}, Cr^{(i,j)}), (Cb^{min}, Cr^{min})\| \\
D_2 &= \|(Cb^{(i,j)}, Cr^{(i,j)}), (Cb^{min}, Cr^{max})\| \\
D_3 &= \|(Cb^{(i,j)}, Cr^{(i,j)}), (Cb^{max}, Cr^{min})\| \\
D_4 &= \|(Cb^{(i,j)}, Cr^{(i,j)}), (Cb^{max}, Cr^{max})\|
\end{aligned} \tag{2}$$

Step 1: Comparison of the Euclidean distances with a proper threshold. If at least one distance is less than the threshold value, then the pixel belongs to the hand region. The proper threshold value is taken equal to 18.

In conclusion, the color segmentation rules are summarized by the following conditions:

$$\begin{cases} (Cb^{(i,j)} \in R_{Cb}) \cap (Cr^{(i,j)} \in R_{Cr}) \Rightarrow (i,j) \in \text{hand} \\ \qquad\qquad\qquad \cup \\ \mathrm{D}_1 \cup D_2 \cup D_3 \cup D_4 \leqslant Threshold \Rightarrow (i,j) \in \text{hand} \end{cases} \tag{3}$$

The output image of the color segmentation process is considered as binary. As illustrated in Fig. 4 the hand region, that is the region of interest, turns black and the background white. The hand region is normalized to certain dimensions so that the system becomes invariant to the hand's size. It is worth to underline also that the segmentation results are very good (almost noiseless) without further processing (e.g. filtering) of the image. In particular, the technique was tested by a set of 180 input images and the rate of successful segmentation was 99.46%.

2.2. Approximation of the hand's morphology

The aim of this stage of the hand recognition process is the approximation of the hand's morphology. This is accomplished by applying the SGONG neural network (Atsalakis and Papamarkos, 2005a, b, 2006; Atsalakis et al., 2005) on the segmented (binary) image.

2.2.1. Self-growing and organized neural gas

The SGONG is an unsupervised neural classifier. It achieves clustering of the input data, so as the distance of the data within the same class (intra-cluster variance) is small and the distance of the data 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) (Kohonen, 1990, 1997) and the Growing Neural Gas (GNG) (Fritzke, 1994, 1995) neural classifiers according to which, the learning rate and the radius of the neighborhood domain of neurons is monotonically decreased during the training procedure. Furthermore, at the end of each epoch of the SGONG classifier, three criteria that improve the growing and the convergence of the network are applied. This is 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 (See Experiment 1) and consequently the shape of the hand.

The SGONG consists of two layers, i.e. the input and the output layer. It has the following main characteristics:

- Is faster than the Kohonen SOFM as the growing mechanism of GNG is used.
- In contrast with GNG classifier, a local counter that influences the learning rate of this neuron and the strength of its connections is defined for each neuron. This local counter depends only on the number of the training vectors that are classified in this neuron.
- The dimensions of the input space and the output lattice of neurons are always identical. Thus, the structure of neurons in the output layer approaches the structure of the input data.
- Criteria are used to ensure fast convergence of the neural network. Also, these criteria permit the detection of isolated classes.

The coordinates of the output neurons are the coordinates of the classes' centers. Each neuron is described by two local parameters related to the training ratio and to the influence by the neighborhood neurons. Both of them decrease from a high to a lower value during a predefined local time in order to gradually minimize the neurons' ability to adapt to the input data. The network begins with only two neurons and it inserts new neurons in order to achieve better data clustering. Its growth is based on the following criteria:

- A neuron is inserted near the one with the greatest contribution to the total classification error, only if the average length

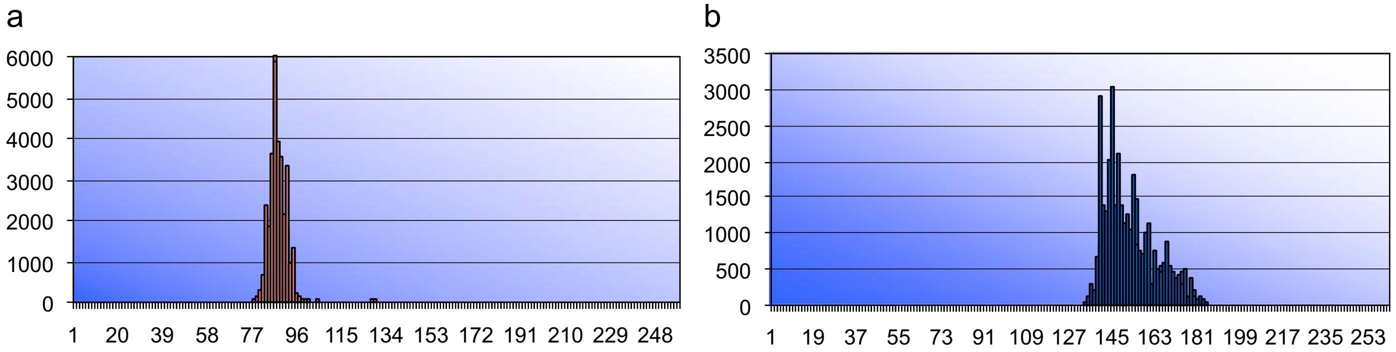


Fig. 3. (a) *Cb* component distribution map and (b) *Cr* component distribution map.

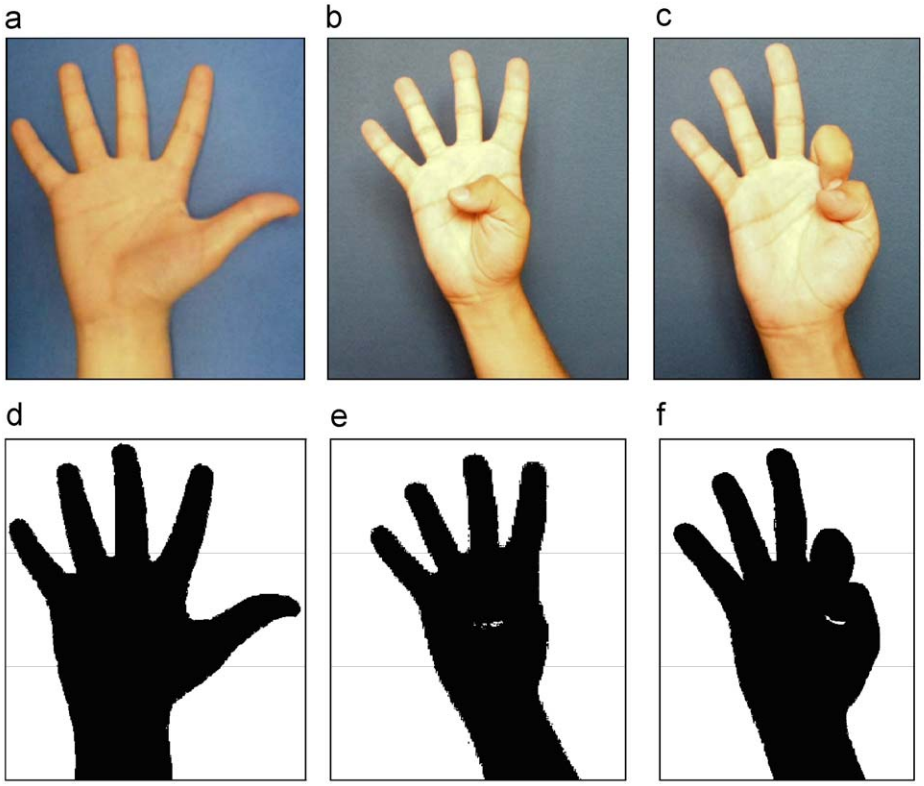


Fig. 4. (a)–(c) Original image and (d)–(f) segmented image.

of its connections with the neighbor neurons is relatively large.

- The connections of the neurons are created dynamically by using the "Competitive Hebbian Learning" method.

The main characteristic of the SGONG is that both neurons and their connections approximate effectively the topology of input data. This is the exact reason for using the specific neural network in this application.

2.2.1.1. The training steps of the SGONG network. The training procedure for the SGONG neural classifier starts by considering first two output neurons ($c=2$). The local counter N_i that expresses the number of vectors that have been classified to the $Neuron_i$ ($i=1,2$), of the created neurons are set to zero. The initial positions of the created output neurons, i.e., the initial values for the weight vectors W_i, $i=1,2$ are initialized by randomly selecting two different vectors from the input space. All the vectors of the training data set X' are circularly used for the training of the SGONG network. The training steps of the SGONG are the following:

Step 1: At the beginning of each epoch the accumulated errors $AE_i^{(1)}, AE_i^{(2)}, \forall i\in[1,c]$, where c is the number of output neurons, are set to zero. The variable $AE_i^{(1)}$ expresses, at the end of each epoch, the quantity of the total quantization error that corresponds to $Neuron_i$, while the variable $AE_i^{(2)}$ represents the increment of the total quantization error that we would have if the $Neuron_i$ was removed.

Step 2: For a given input vector X_k, the first and the second winner neurons $Neuron_{w1}$, $Neuron_{w2}$ are obtained:

$$\text{for } Neuron_{w1}\ ||X_k - W_{w1}|| \leqslant ||X_k - W_i|| \forall i \in [1, c] \tag{4}$$

$$\text{for } Neuron_{w2}\, X_k - W_{w2}|| \leqslant ||X_k - W_i|| \forall i \in [1,\ c] \text{ and } i \neq w1 \tag{5}$$

Step 3: The local variables $AE_i^{(1)}$ and $AE_i^{(2)}$ change their values according to the relations:

$$AE_{w1}^{(1)} = AE_{w1}^{(1)} + ||X'_k - W'_{w1}|| \tag{6}$$

$$AE_{w1}^{(2)} = AE_{w1}^{(2)} + ||X'_k - W'_{w2}|| \tag{7}$$

$$N_{w1} = N_{w1} + 1 \tag{8}$$

where N_{w1} is the number of vectors classified to the neuron $Neuron_{w1}$.

Step 4: If $N_{w1} \leqslant N_{idle}$ (The variable N_{idle} determines the required number of consecutive vectors that should be classified to a class in order to define a well-trained neuron.) then the local learning rates $\varepsilon 1_{w1}$ and $\varepsilon 2_{w1}$ change their values according to Eqs. (9)–(11). Otherwise, the local learning rates have the constant values $\varepsilon 1_{w1} = \varepsilon 1_{min}$ and $\varepsilon 2_{w1} = 0$ and

$$\varepsilon 2_{w1} = \varepsilon 1_{w1}/r_{w1} \tag{9}$$

$$\varepsilon 1_{w1} = \varepsilon 1_{max} + \varepsilon 1_{min} - \varepsilon 1_{min} \left(\frac{\varepsilon 1_{max}}{\varepsilon 1_{min}}\right)^{N_{w1}/N_{idle}} \tag{10}$$

$$r_{w1} = r_{max} + 1 - r_{max} \left(\frac{1}{r_{max}}\right)^{N_{w1}/N_{idle}} \tag{11}$$

The learning rate $\varepsilon 1_i$ is applied to the weights of $Neuron_i$ if this is the winner neuron ($w1=i$), while $\varepsilon 2_i$ is applied to the weights of $Neuron_i$ if this belongs to the neighborhood domain of the winner neuron ($i \in nei(w1)$). The learning rate $\varepsilon 2_i$ is used in order to have soft competitive effects between the output neurons. That is, for each output neuron, it is necessary that the influence from its neighboring neurons to be gradually reduced from a maximum to a minimum value. The values of the learning rates $\varepsilon 1_i$ and $\varepsilon 2_i$ are not constant but they are reduced according to the local counter N_i. Doing this, the potential ability of moving neuron i towards an input vector (plasticity) is reduced by time. Both learning rates change their values from maximum to minimum in a period, which is defined by the N_{idle} parameter. The variable r_{wi} initially takes its minimum value $r_{min}=1$ and in a period, defined also by the N_{idle} parameter, reaches its maximum value r_{max}.

Step 5: In accordance to the Kohonen SOFM, the weight vector of the winner neuron $Neuron_{w1}$ and the weight vectors of its neighboring neurons $Neuron_m$, $m \in nei(w1)$, are adapted according to the following relations:

$$W'_{w1} = W'_{w1} + \varepsilon 1_{w1}(X'_k - W'_{w1}) \tag{12}$$

$$W'_m = W'_m + \varepsilon 2_m(X'_k - W'_m),\ \forall m \in nei(w1) \tag{13}$$

Step 6: With regard to generation of lateral connections, SGONG employs the following strategy. The Competitive Hebbian Rule is applied in order to create or remove connections between neurons. As soon as the neurons $Neuron_{w1}$ and $Neuron_{w2}$ are detected, the connection between them is created or is refreshed. That is

$$s_{w1,w2} = 0 \tag{14}$$

With the purpose of removing superfluous lateral connections, the age of all connections emanating from $Neuron_{w1}$, except the connection with $Neuron_{w2}$, is increased by one:

$$s_{w1,m} = s_{w1,m} + 1,\ \forall\ m \in nei(w1) \text{ with } m \neq w2 \tag{15}$$

where

$$s_{i,j} = s_{j,i} \geqslant -1, \forall i,j \in [1,c] \text{ with } i \neq j \tag{16}$$

If the connection between $Neuron_i$ and $Neuron_j$ exists then $s_{i,j} \geqslant 0$, otherwise $s_{i,j} = -1$. The expressions $s_{i,j}$ and $s_{j,i}$ are considered as equal. If the connection $s_{i,j}$ exists, the positive value of quantity $s_{i,j}$ expresses the age of the lateral synapse.

Step 7: At the end of each epoch it is examined if all neurons are in *idle state*, or equivalently, if all the local counters N_i, $\forall i \in [1,c]$ are greater than the predefined value N_{idle} and the neurons are considered well trained. In this case, the training procedure stops and the convergence of SGONG network is assumed. The number of input vectors needed for a neuron to reach the *idle state* influences the convergence speed. If the training procedure continues, the lateral connections between neurons with age greater than the maximum value α are removed. Due to dynamic generation or removal of lateral connections, the neighborhood domain of each neuron changes in time in order to include neurons that are topologically adjacent.

Step 8: Also, three criteria that modify the number of the output neurons c and make the proposed neural network to become self-growing are applied. These criteria are applied in the following order:

- A class (neuron) is removed if for a predefined consecutive number of epochs, none of the training samples has been classified in this class.
- A new class (neuron) is added near the class with the maximum contribution to the total quantization error (with the maximum $AE^{(1)}$), if the average distance of its vectors from neighboring classes is greater than a predefined value. This value is expressed as a percentage of the average distance between all classes.
- The class (neuron) with the minimum average distance of its vectors from neighboring classes is removed if this quantity is less than a predefined value. This value is expressed as a percentage of the average distance between all classes.

In order to make the network convergence faster it can be defined not to apply the above criteria when the total number of epochs is above a predefined value. This has as a result the rapid passing of all neurons to the *idle state* and therefore the finalizing of the training procedure. After the training procedure, the denormalized vectors W_i, $i = 1, 2, \ldots, c$ express the centers of the final classes, i.e. the coordinates of the output neurons.

A detailed description of SGONG can be found in Atsalakis and Papamarkos (2005a, b, 2006) and Atsalakis et al. (2005) while its implementation can be found in http://www.papamarkos.gr/uploaded-files/Papaparkos/demos/sgong_demo.htm.

2.2.2. *Application of the self-growing and self-organized neural gas network*

In the proposed method the input data of the SGONG are the coordinates of random samples of the black/hand pixels. Let $X_k = (i,j)$ be the kth input vector, where (i,j) are the coordinates of a randomly selected black pixel and $k \in [1, N_{iv}]$. The number N_{iv} of the input vectors that are used for the training process is chosen to be approximately 5% of the black pixels, in order to achieve satisfactory approximation of the hand shape and fast time convergence. If $N_{iv} \ll 5\%$ the SGONG describes less adequately the hand and if $N_{iv} \gg 5\%$, it converges slowly to a grid of output neurons similar to the one created by using $N_{iv} \simeq 5\%$.

During the training, the network grows gradually on the hand region and a structure of neurons and their connections is finally created. The output neurons' coordinates are calculated by using Eqs. (12) and (13) and the criteria described in Step 8 of the training process. These coordinates correspond to pixels of the black segment. Let $W_p = (i,j)$ be the pth weight vector, i.e. (i,j) the coordinates of the pth output neuron, and let s be the 2D array that describes the connections between the output neurons. Specifically, if $s_{pq} = -1$ then the output neuron p is not connected with the output neuron q. If $s_{pq} > 0$, then there is a connection between the output neurons p and q.

At the end of the training process, the SGONG defines approximately 80 classes on the hand region. It is obvious

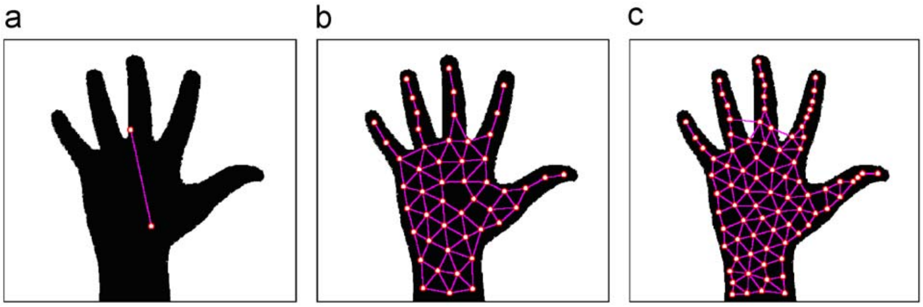


Fig. 5. Growth of the SGONG network: (a) starting point—2 neurons, (b) growing stage—45 neurons and (c) final output grid of 83 neurons.

however, as shown in Fig. 5, that the shape of the hand could be described by using fewer output neurons.

A smaller set of output neurons is desirable, because it results in faster processing and thus faster finger features extraction. Therefore, a sufficient number of final classes is used as a threshold parameter of the SGONG's training process. The final number of the output neurons should satisfy the following criteria:

- Each finger should be described by a small number of neurons.
- The grid of neurons should approximate successfully the hand contour.

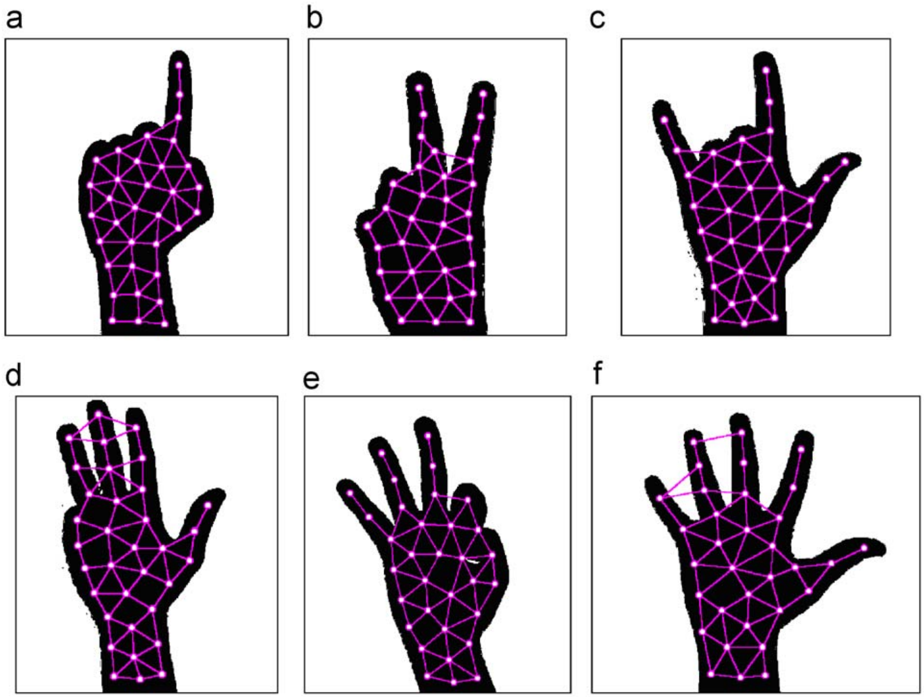


Fig. 6. Final output grid of 33 neurons of various input images.

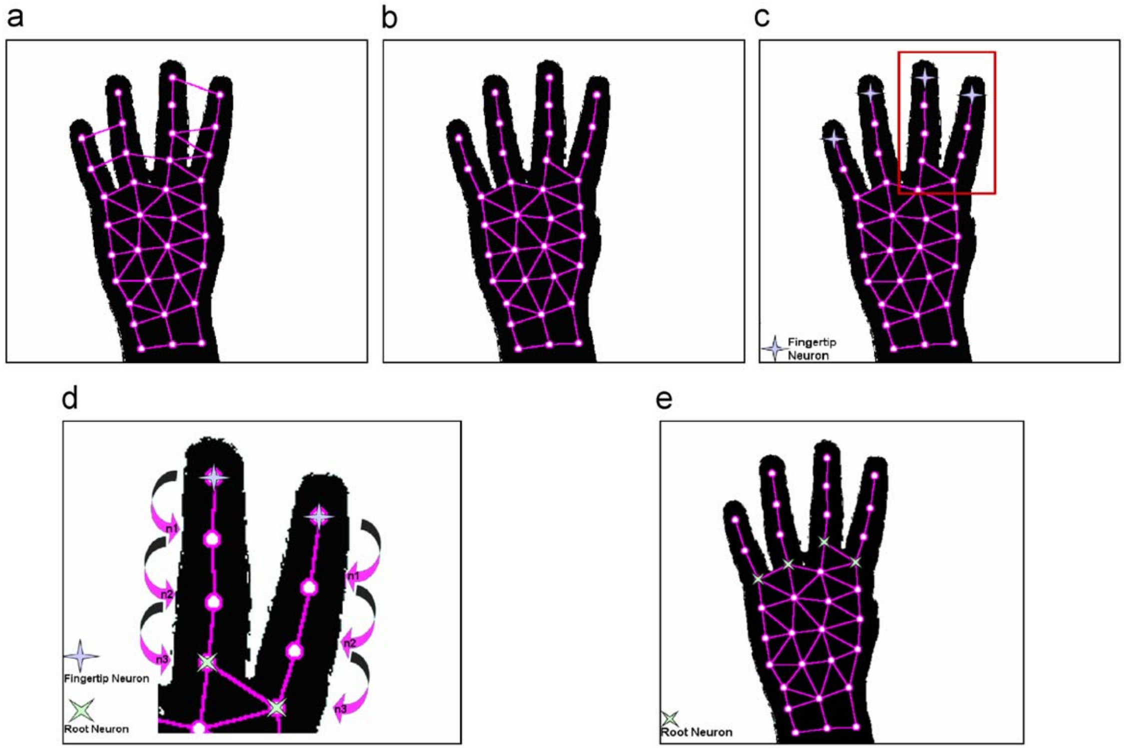


Fig. 7. (a) Grid of output neurons after the application of the SGONG, (b) removal of the connection that go through the background, (c) determination of the fingertip neurons, (d) successive determination of the finger neurons and (e) determination of the root neurons.

- The grid of neurons should approximate successfully the palm region.

After testing, we have found that the proper number of neurons that satisfies these rules is 33. The satisfactory approximation of the morphology of the hand using 33 output neurons is shown in Fig. 6.

Finally, it is worth to underline that the output data of the network is the array of the neurons' coordinates W_p and the 2D array of the neurons' connections s_{pq}. Based on this information, important finger features are extracted.

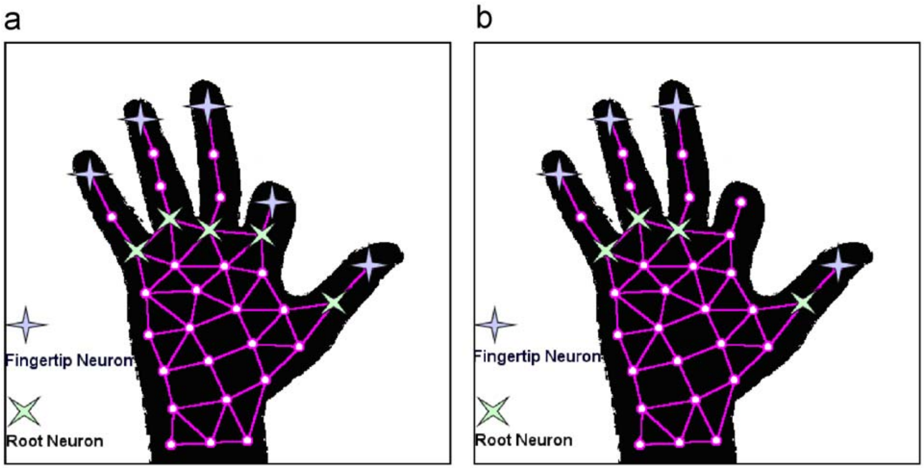


Fig. 8. (a) False finger detection and (b) correct finger detection, after applying the mean finger length comparison check.

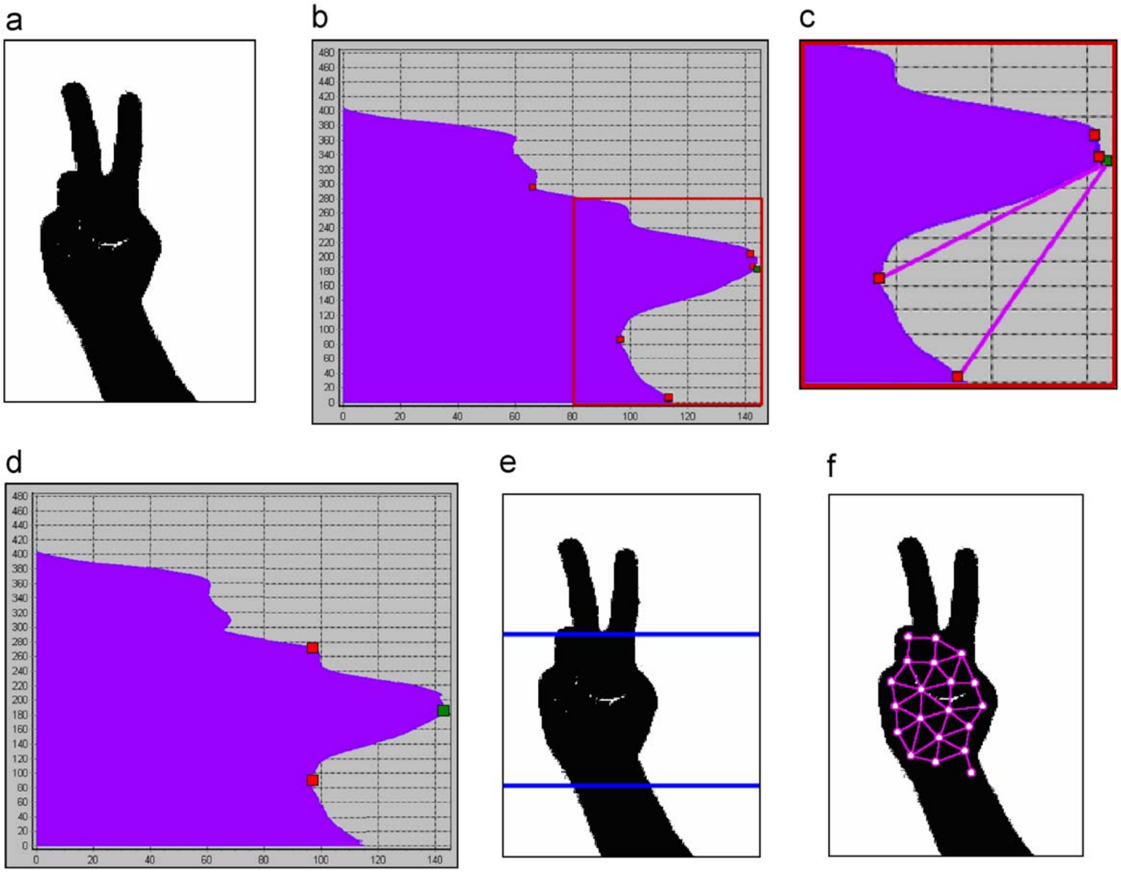


Fig. 9. (a) Binary image, (b) horizontal projection of the binary image (green point denotes the global maximum and red points denote the local minima of the projection), (c) lines segments connecting the global maximum and the local minima, (d) coordinates of j_{lower} (95, 90) and coordinates of j_{upper} (95, 275), (e) defined palm region and (f) palm region neurons. For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.

2.3. Finger identification

The recognition of static hand gestures can be implemented by finger identification. Therefore the proposed method extracts robust features that describe successfully the properties of the fingers. The features are invariant to the hand's morphology as well as to its slope and size. Moreover, the features' values are discrete for every type of finger and exploit efficiently the morphologic information of the grid of the output neurons. The finger identification process consists of the following stages:

Stage 1: Determination of the number of the raised fingers.

Stage 2: Extraction of hand shape characteristics.

Stage 3: Extraction of finger features.

An analysis of the above stages follows.

2.3.1. Determination of the number of the raised fingers

The aim of this stage is to determine the number of the raised fingers as well as the coordinates of the neurons that represent them. The most important finger neurons are: (a) the neurons that correspond to the fingertips (fingertip neurons) and (b) the neurons that describe the fingers' lower limit (root neurons). The determination of the raised fingers is accomplished by locating the fingertip neurons, which are also used as a starting point for the detection of the rest of the finger neurons.

Observations of the structure of the output neurons' grid lead to the conclusion that fingertip neurons are connected to neighborhood neurons by only two types of connections: (i) connections that go through the background and (ii) connections that belong exclusively to the hand region. The crucial point is that fingertip neurons use only one connection of the second type. Based on this conclusion, the process of the determination of the number of fingers is as follows:

Step 1: Remove all the connections that go through the background (Fig. 7(b)).

Step 2: Find the neurons that have only one connection. As indicated in Fig. 7(c), these neurons are the fingertips.

Step 3: Starting from the fingertip neurons find successively the neighbor neurons. Stop when a neuron with more than two connections is found. This is the finger's last neuron (root neuron) (Fig. 7(d) and (e)).

In special cases, the above algorithm leads to false conclusions. For example, as shown in Fig. 8(a), the algorithm detects five fingertips, although the gesture consists of only four fingers. This type of error can be avoided by comparing every finger's length (i.e. the fingertip and root neuron distance) with the mean fingers' length. If a finger's length differs significantly from the mean value then it is not considered to be a finger. The results of this check are shown in Fig. 8(b).

2.3.2. Extraction of hand shape characteristics

The morphology of the hand affects and changes the values of the fingers' features. Therefore, it is necessary to specify the fundamental characteristics of the hand's shape before proceeding to the feature extraction.

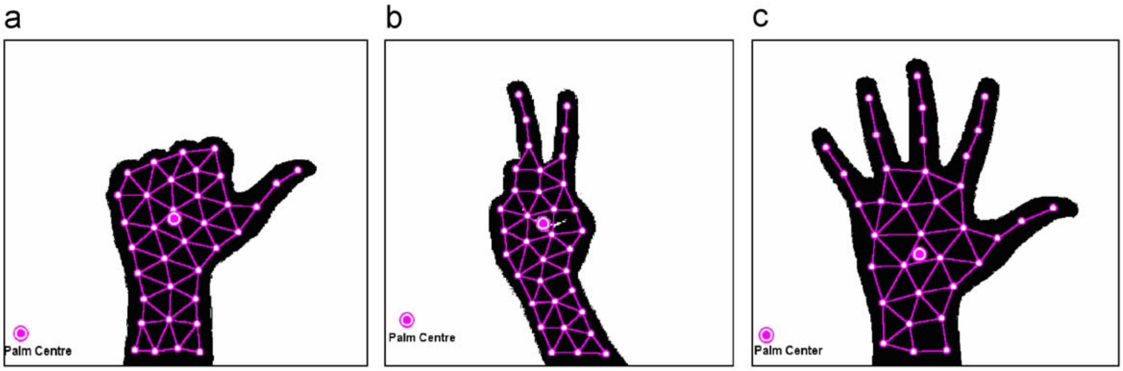


Fig. 10. Palm centers of various input images.

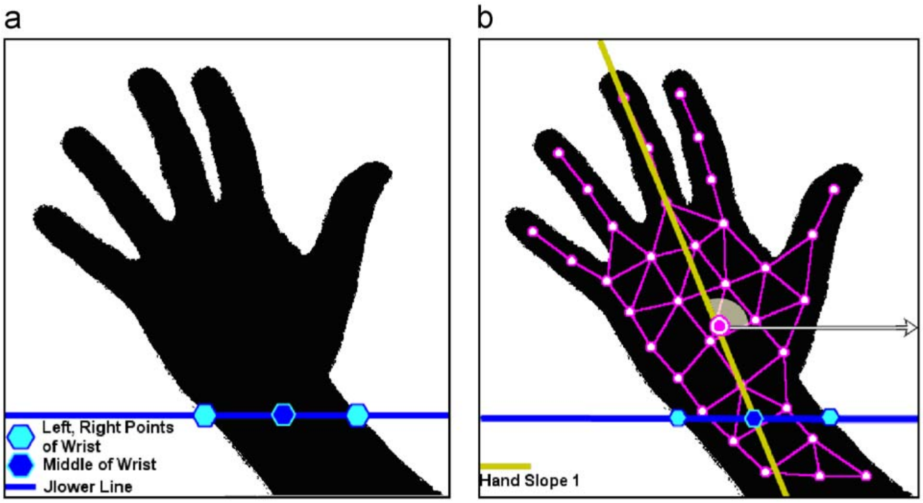


Fig. 11. (a) Location of the middle of the wrist and (b) hand slope based on the first technique.

2.3.2.1. Palm region. Many input images include redundant information, such as the presence of a part of the arm. This redundant information could reduce the accuracy of the extraction techniques and lead to false conclusions. Therefore, it is important to locate the hand region that describes most effectively and accurately the morphology properties, i.e. the palm.

The algorithm of finding the palm region is based on the observation that the arm is thinner than the palm. Thus, a local minimum should appear at the horizontal projection of the binary image. This minimum defines the limits of the palm region. This procedure is as follows:

Step 1: Create the horizontal projection of the binary image $H[j]$, $j\in[1, \text{Image Height}]$. Apply a mean filter on the horizontal projection, in order to reduce the local variance for every j.

Step 2: Find the global maximum $H[j_{GlobalMax}]$ and each one of the local minima $H[j_{min}]$ of $H[j]$ (Fig. 9(b)).

Step 3: Calculate the slope of the line segments connecting the global maximum and each one of the local minima that satisfies

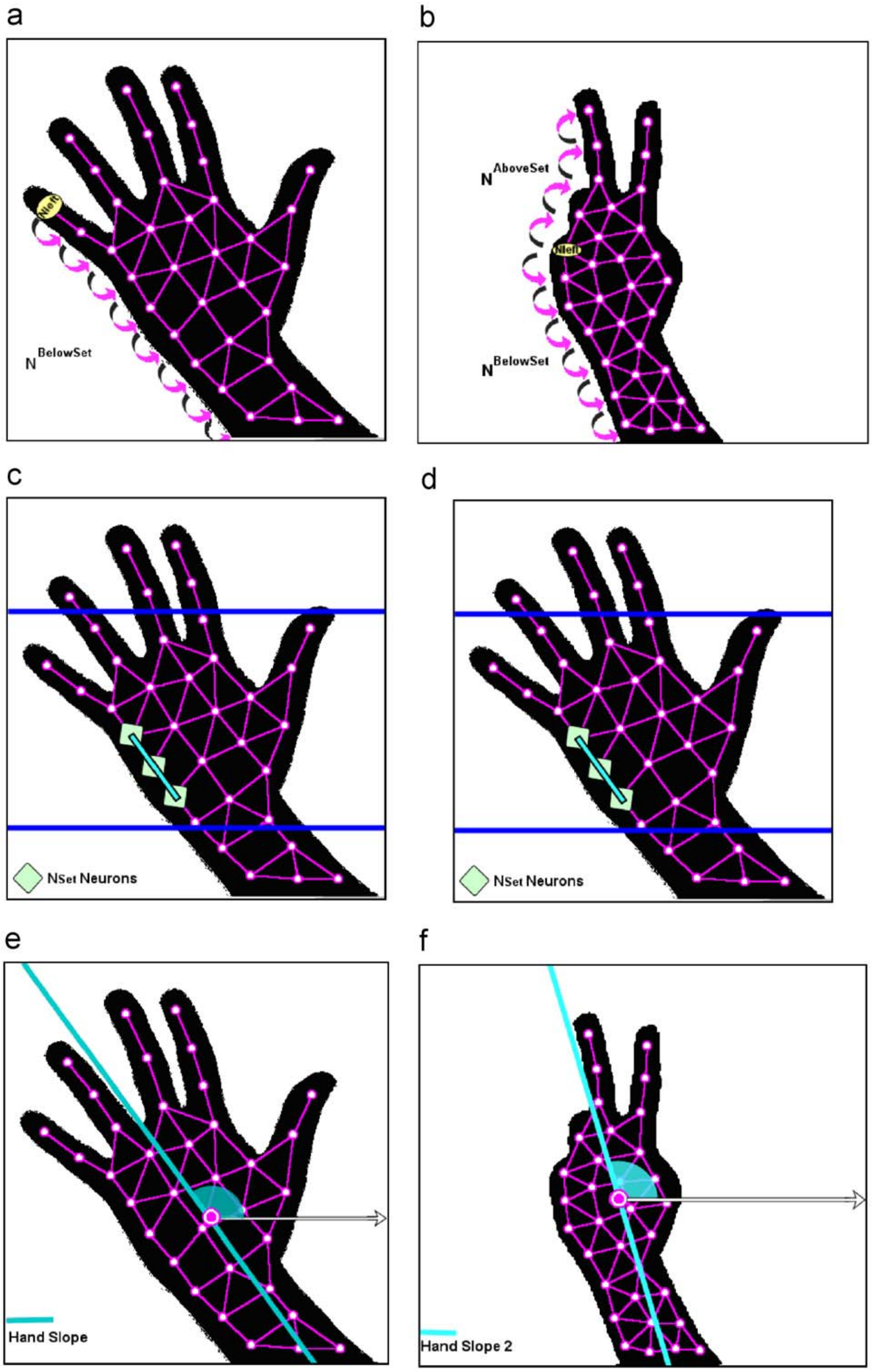


Fig. 12. (a) and (b) Location of the $N_{AboveSet}$ and $N_{BelowSet}$ neurons, (c) and (d) final N_{Set} neurons and (e) and (f) hand slope according to the second technique.

the condition $j_{min} < j_{GlobalMax}$:

$$Slope = \frac{H[j_{GlobalMax}] - H[j_{min}]}{j_{GlobalMax} - j_{min}} \tag{17}$$

The minimum $H[j_{min}]$ that corresponds to the greatest of these slopes is denoted as j_{lower} and defines the lower limit of the palm region only if its distance from the maximum is greater than a threshold value equal to ImageHeight/6 (Fig. 9(c)).

Step 4: The point that defines the upper limit of the palm region is denoted as j_{upper} and is obtained by the following relation (Fig. 9(d)):

$$H[j_{upper}] \leqslant H[j_{lower}] \text{ and } j_{upper} > j_{GlobalMax} > j_{lower} \tag{18}$$

Step 5: The palm region is defined by the lines j_{lower} and j_{upper}. In order to achieve greater accuracy, the finger neurons are not included in the set of palm region neurons (Fig. 9(e) and (f)).

2.3.2.2. Palm center. The coordinates of the center of the palm are taken equal to the gravity center of the coordinates of the neurons that belong to the palm region. Let (x_i, y_i) be the coordinates of the N palm neurons. Then the coordinates of the palm center are defined by according to the following equation:

$$x_{pc} = \frac{1}{N}\sum_{i=1}^{N} x_i \text{ and } y_{pc} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{19}$$

Fig. 10 shows three examples of the determination of the palm center.

2.3.2.3. Hand slope. Despite the roughly vertical direction of the arm, the slope of the hand varies. This fact should be taken into consideration because it affects the accuracy of the finger features extraction, and consequently, the efficiency of the identification process. The recognition results depend greatly on the correct calculation of the hand slope. In order to achieve more accurate results, the estimation of the hand slope is based on the combination of two different techniques.

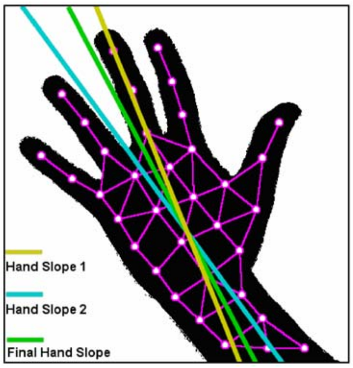


Fig. 13. Final hand slope that takes under consideration both techniques.

According to the first technique, the hand slope is equal to the angle of the line segment connecting the palm center and the middle of the wrist with the horizontal axis. The steps of this algorithm are the following:

Step 1: Define the middle of the wrist (x_{wrist}, y_{wrist}). By using the line that corresponds to the j_{lower} of the palm region, one can locate the leftmost point of the wrist as the first black pixel that belongs to the j_{lower} line and the rightmost point of the wrist as the last black pixel (Fig. 11(a)).

Step 2: The slope (Fig. 11(b)) of the line segment that connects the middle of the wrist and the palm center is given by

$$HandSlope^1 = tan^{-1}\left(\frac{y_{pc} - y_{wrist}}{x_{pc} - x_{wrist}}\right) \tag{20}$$

According to the second technique, the hand slope is estimated by the angle of the left side of the palm. The technique consists of the following steps:

Step 1: Find the neuron N_{Left}, which belongs to the palm region and has the smallest horizontal coordinate.

Step 2: Obtain the set of palm neurons $N_{AboveSet}$ that belong to the upper left boundary of the neurons grid. To do this, and for each neuron, starting from the N_{Left}, we obtain the neighborhood neuron which has, simultaneously, the highest vertical and the lowest horizontal coordinates (Fig. 12(a)).

Step 3: Obtain the set of palm neurons $N_{BelowSet}$ that belong to the lower left boundary of the neurons grid. To do this, and for each neuron, starting from the N_{Left}, we obtain the neighborhood neuron which has, simultaneously, the lowest vertical and horizontal coordinates (Fig. 12(b)).

Step 4: Remove from the N_{Set} $(N_{Set} = N_{AboveSet} \cup N_{BelowSet})$ the finger neurons and the neurons that do not belong to the palm region.

Step 5: Calculate the difference of slopes of the line segments that connect two successive neurons. Remove from the N_{Set} the neurons whose slope differs from the previous slope more than a predefined threshold.

Step 6: The first and the final neurons of the set N_{Set} define the hand's slope (Fig. 12(c) and (d)).

The final estimation of the hand slope is based on both techniques and is calculated by the equation:

$$HandSlope = 0.6HandSlope^1 + 0.4HandSlope^2 \tag{21}$$

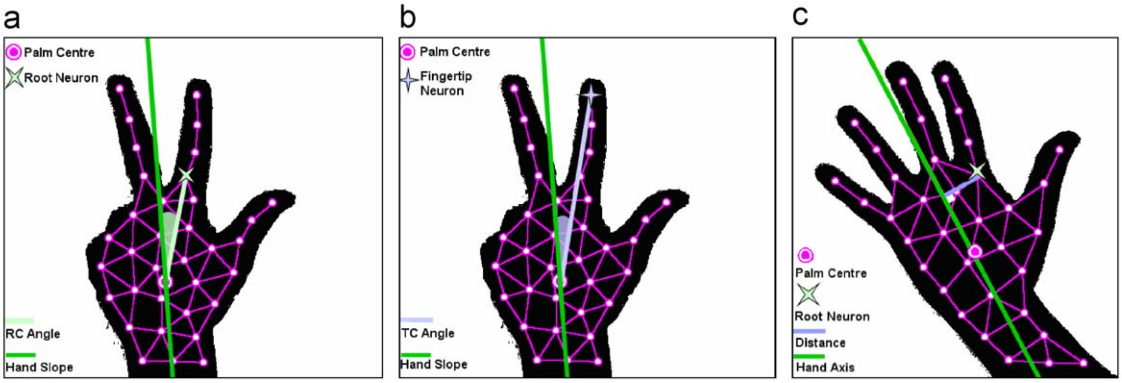
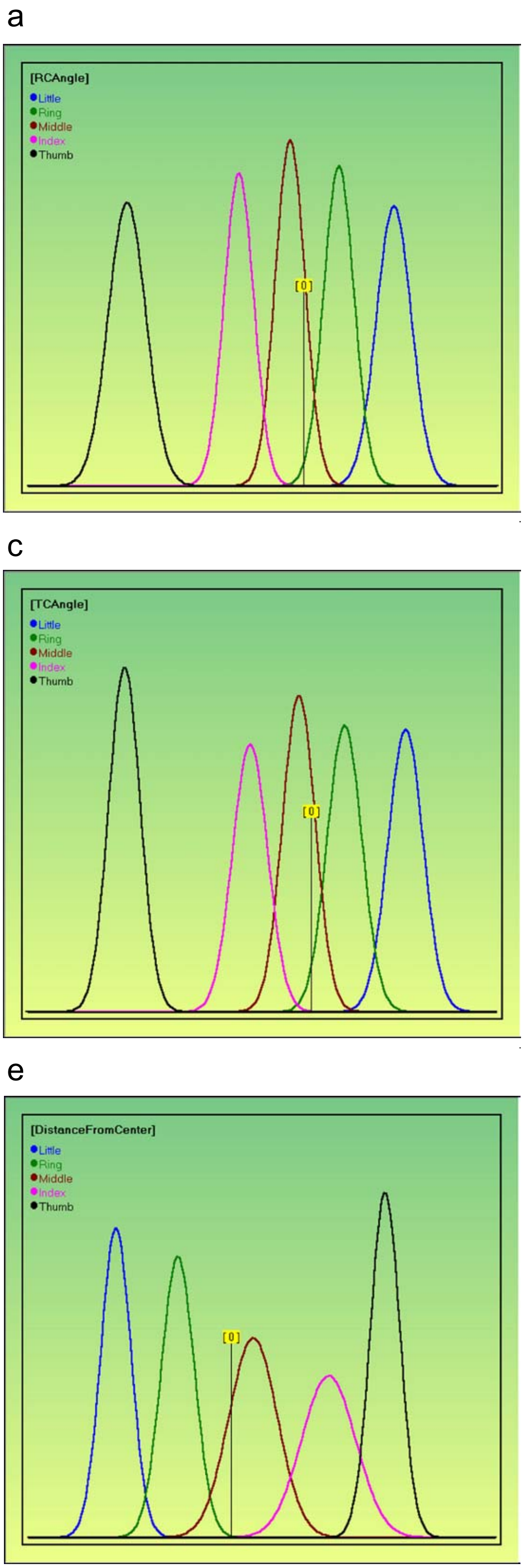


Fig. 14. (a) RC Angle, (b) TC Angle and (c) distance from the palm center.

As shown in Fig. 13, the hand slope is successfully approximated. Let Hand Slope Line (HSL) be the line that passes through the palm center and forms an angle with the horizontal axis equal to the hand slope. The hand slope is considered as a reference angle and is used in order to improve the finger features' extraction techniques.



b

Class	Mean Value	Standard Deviation
Little	40.99	7.91
Ring	16.03	6.92
Middle	-6.01	6.7
Index	-29.26	6.68
Thumb	-79.89	8.52

d

Class	Mean Value	Standard Deviation
Little	34.12	6.36
Ring	12.03	6.27
Middle	-4.45	5.92
Index	-22.06	6.31
Thumb	-67.52	5.68

f

Class	Mean Value	Standard Deviation
Little	-51.52	6.73
Ring	-23.71	7.43
Middle	10.14	10.92
Index	44.06	12.16
Thumb	69.45	6.6

Fig. 15. Features distributions (a) and (b) RC Angle, (c) and (d) TC Angle and (e) and (f) distance from the center.

2.3.3. *Extraction of finger features*

The extracted features describe morphologic and geometric properties of the fingers. The method proposes the extraction of three features.

2.3.3.1. Finger angles. A geometric feature that individualizes the fingers is their, relative to the hand slope, angles. The two different types of angles are the following:

- *RC Angle*: It is an angle formed by the HSL and the line that joints the root neuron and the hand center (Fig. 14(a)). This angle provides the most discrete values for each finger and thus is valuable for the recognition:

$$R\hat{C} = HandSlope - tan^{-1}\left(\frac{y_{root} - y_{pc}}{x_{root} - x_{pc}}\right) \tag{22}$$

- *TC Angle*: It is an angle formed by the HSL and the line that joints the fingertip neuron and the hand center (Fig. 14(b)). It is used directly for the finger identification process:

$$T\hat{C} = HandSlope - tan^{-1}\left(\frac{y_{fingertip} - y_{pc}}{x_{fingertip} - x_{pc}}\right) \tag{23}$$

2.3.3.2. Distance from the palm center. A powerful feature for the identification process is the vertical distance of the finger's root neuron from the line passing through the palm center and having the same slope as the HSL. An example is illustrated in Fig. 14(c). 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 distance between the leftmost and the rightmost neuron of the palm region.

2.4. *Recognition process*

The final stage of the proposed method is, of course, the recognition of the hand gesture. The recognition process is based on the choice of the most probable finger combination of a set of feasible gestures. This is accomplished by classifying the raised fingers into five classes (thumb, index, middle, ring, little) according to their features. The classification depends on the probabilities of a finger to belong to the above classes. The probabilities derive from the features' distributions. Therefore, the recognition process consists of three stages:

Stage 1: The off-line calculation of the features' distributions.
Stage 2: The likelihood-based classification.
Stage 3: Final classification.

2.4.1. *Calculation of features' distributions*

The finger features are naturally occurring features. Hence a Gaussian distribution could model them successfully. Their distributions are calculated by using a training set of 100 images from different people. The following process is carried out off-line and is regarded as the training process of the proposed hand recognition system.

If f_i is the ith feature ($i \in [1,3]$), then its Gaussian distributions for every class (finger class) c_j ($j \in [1,5]$) are given by the relation:

$$p_{fi}^{c_j}(x) = \frac{e^{-(x-m_{f_i}^{c_j})^2/(2\sigma_{fi}^{cj})^2}}{\sigma_{fi}^{cj}\sqrt{2\pi}} \tag{24}$$

where $j = 1, \ldots, 5$, m_{fi}^{cj} is the mean value and σ_{fi}^{cj} is the standard deviation of the f_i feature of the c_j class. The Gaussian distributions of the above features are shown in Fig. 15. As it can be observed, the five classes are well defined and well discriminated.

Table 1
Values of finger features and RP_{cj} possibilities calculated for the input image shown in Fig. 16(a).

Finger	Features' values		Feature likelihood/class				
			Little	Ring	Middle	Index	Thumb
1.	TC	23.18	0.0143	0.0131	0	0	0
	RC	33.69	0.0329	0.0022	0	0	0
	Distance	−48.9	0.0550	0	0	0	0
	Sum		0.1022	0.0153	0	0	0
2.	TC	4.94	0	0.0336	0.0192	0	0
	RC	10.15	0	0.0402	0.0032	0	0
	Distance	−17.3	0	0.0370	0.0016	0	0
	Sum		0	0.1108	0.0240	0	0
3.	TC	−13.2	0	0	0.0226	0.0208	0
	RC	−16.6	0	0	0.0171	0.0099	0
	Distance	24	0	0	0.0163	0.0084	0
	Sum		0	0	0.056	0.0391	0
4.	TC	−30.3	0	0	0	0.0245	0
	RC	−39.9	0	0	0	0.0168	0
	Distance	61.65	0	0	0	0.0115	0.0301
	Sum		0	0	0	0.0528	0.0301

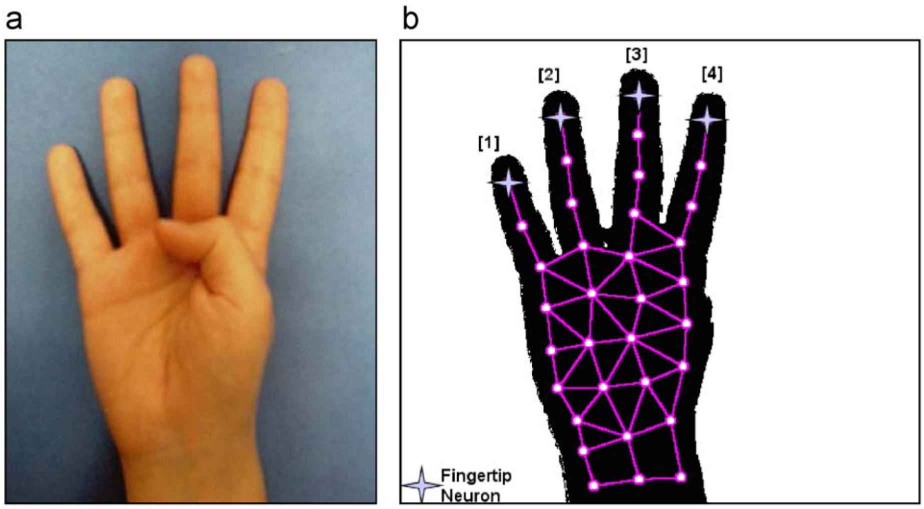


Fig. 16. (a) Input image and (b) numbering of fingers.

2.4.2. Likelihood-based classification

The first step of the classification process is the calculation of the likelihood RPc_j of a raised finger to belong to each one of the five classes. Let $x0$ be the value of the ith feature f_i. Calculate the likelihood $p_{fi}^{cj}(x0)$ for $i\in[1,3]$ and $j\in[1,5]$. The requested likelihood is the sum of the likelihoods of all the features for each class and is calculated according to the following equation:

$$RPcj = \sum_{i=1}^{3} p_{f_i}^{c_j} \tag{25}$$

For example, let Fig. 16(a) be the input image. As analyzed previously, the image is processed (segmentation and application of the SGONG), the finger features are extracted and finally the possibilities of every raised finger to belong to each one of the five classes are calculated. The raised fingers are numbered as shown in Fig. 16(b). Table 1 indicates the features' values of every raised finger of Fig. 16(b), as well as the likelihood RPc_j.

For example, finger No. 2 belongs to the following classes in order of higher possibility: Ring, Middle. It is worth to underline that even if the value of a feature likelihood $p_{fi}^{cj}(x0)$ leads to false classification, the sum of the likelihoods RPc_j will eliminate the error and lead to correct classification. For instance, finger No. 4 is classified falsely as Thumb according to the Distance value. The sum RPc_j, however, classifies it correctly as Index.

The above process has the disadvantage that two fingers may be classified to the same class. Therefore, it is used only as a starting point of the final recognition process.

Table 2
Set of feasible gestures when the number of raised finger is 4.

Little	Ring	Middle	Index	Thumb	Sum
x	x	x	x	–	0.3218
x	x	x	–	x	0.2991
x	x	–	x	x	0.2822
x	–	x	x	x	0.1954
–	x	x	x	x	0.1085

The Sum defines the possibility of each gesture to correspond to the input image of Fig. 16(a).

2.4.3. Final classification

The hand gesture recognition is accomplished by choosing the most probable finger combination. Firstly, the algorithm defines all the feasible gestures by calculating the combination of the five classes to the number of raised fingers:

$$\binom{5}{N} = \frac{5!}{N!(5-N)!} \tag{26}$$

where N is the number of raised fingers. For example, Table 2 presents the feasible gestures when $N=4$. The empty classes (fingers that are not raised) are denoted by "–", whereas the non-empty classes by "x".

Considering the order of classes as it appears at Table 2, the non-empty classes are numbered from left to right. Then, for every one of the feasible gestures the sum of the likelihood of the ith finger to belong to the ith non-empty class is calculated. For example, the possibility of the first gesture of Table 2 is calculated by summing the likelihood of finger No. 1 to be Little, finger No. 2 to be Ring, finger No. 3 to be Middle and finger No. 4 to be Index. As shown in Table 2, this gesture is the most probable and thus it is considered to correspond to the input image of Fig. 16(a).

3. Experimental results

The hand gesture recognition system, which was implemented in Delphi, was tested by using hand images from different people

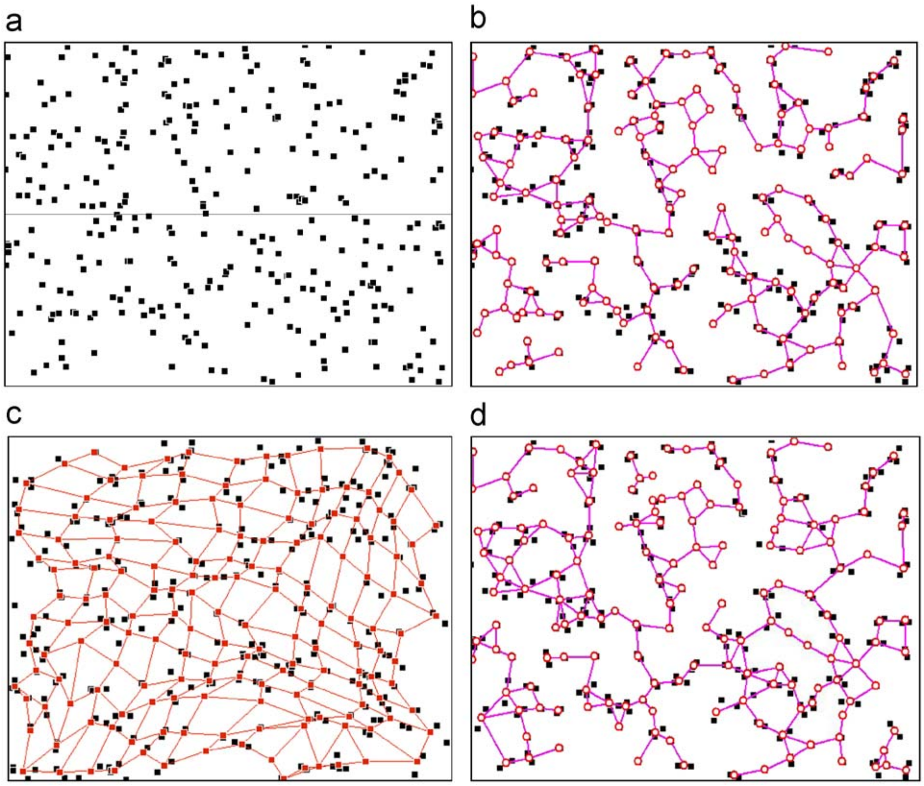


Fig. 17. (a) Input image; final output of (b) SGONG after 286 epochs, (c) Kohonen SOFM after 372 epochs and (d) GNG after 299 epochs.

with varying morphology, slope and size. The conclusions drawn concern the robustness of the features as well as the recognition rate of the system.

3.1. Experiment 1

The innovation of the proposed method is the use of the SGONG neural network in order to approximate the hand's morphology. SGONG combines the advantages of both the Kohonen SOFM and the GNG neural network. Its main advantage is that it can adaptively determine the final number of neurons and thus it can capture the feature space effectively. The following experiment will show that the SGONG network converges faster compared to the Kohonen SOFM and GNG networks, and that achieves effective description of the structure of the input data.

Consider the image of Fig. 17(a) as the input space (i.e. the coordinates of the black pixels are the input vectors of the network). Application of the SGONG on the image leads to the determination of 163 output neurons. The SGONG converges after 286 epochs and, as Fig. 17(b) shows, describes the input space very well. As for as the Kohonen SOFM is concerned, the grid of output neurons is determined to be 13 × 13. It converges (Fig. 17(c)) slower than the SGONG after 372 epochs. The GNG neural network uses as an input parameter the final number of 163 output neurons and it converges (Fig. 17(d)) after 299 epochs. It is worth to underline that the GNG and mainly the SGONG neural network describe efficiently the isolated classes contrary to the Kohonen SOFM, which preserves its initial neighbor neuron connections.

Fig. 18(a)–(f) show stages of the growing procedure of the three neural classifiers.

Table 3
Finger recognition rate for every feature.

Feature	RC angle (%)	TC angle (%)	Distance from center (%)
Recognition rate	94.05	90.36	92.22

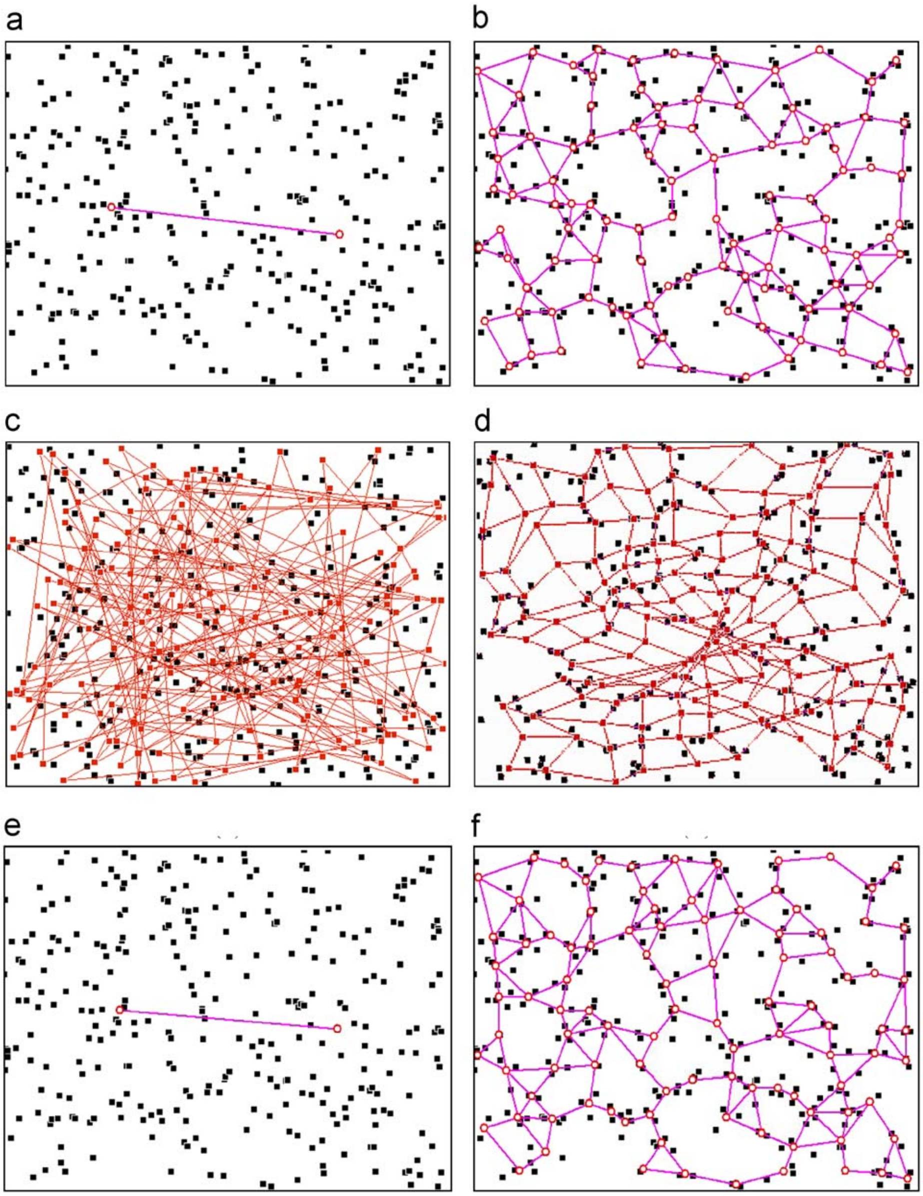
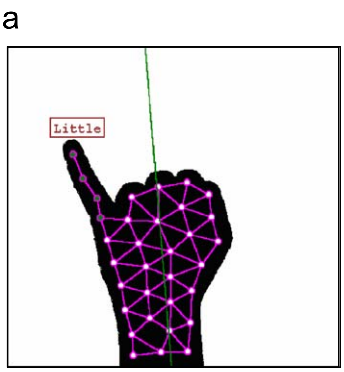


Fig. 18. Starting point of: (a) SGONG, (c) Kohonen SOFM, (e) GNG; Intermmediate stage of: (b) SGONG (100 neurons–98 epochs), (d) Kohonen SOFM (169 neurons–170 epochs) and (f) GNG (100 neurons–98 epochs).

3.2. *Experiment 2*

The goal of this experiment is to study the extracted features' effectiveness, because it plays a significant role in the outcome of the recognition process. The effectiveness of a feature is associated with the value of the finger recognition rate achieved. The higher the recognition rate, the more effective the feature. Using the set of 503 fingers of the 180 input gestures the recognition rates for every feature are shown in Table 3.

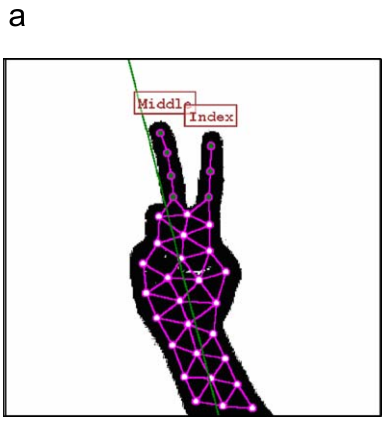
The above finger recognition rates are justified by taking into consideration the features' distributions shown in Fig. 15. The lowest recognition rate is the one of TC angles, because as shown in Fig. 15(b) it has the less discriminated distribution. As far as the feature distance from the center of the palm is concerned, the class of Index is not well separated from the class of the Thumb.



b

Little	Ring	Middle	Index	Thumb	Score
x	-	-	-	-	28.597
-	x	-	-	-	0.016
-	-	x	-	-	0
-	-	-	x	-	0
-	-	-	-	x	0

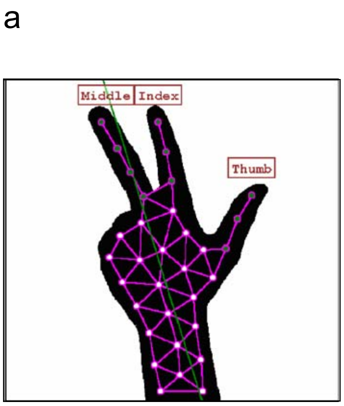
Fig. 19. (a) Recognition of a gesture with one raised finger and (b) possibilities of every one feasible gesture.



b

Little	Ring	Middle	Index	Thumb	Score
-	-	x	x	-	58.396
-	x	-	x	-	31.97
x	-	-	x	-	29.816
-	-	x	-	x	28.581
-	x	x	-	-	2.627
-	x	-	-	x	2.156
x	-	x	-	-	0.472
-	-	-	x	x	0.066
x	-	-	-	x	0.001
x	x	-	-	-	0

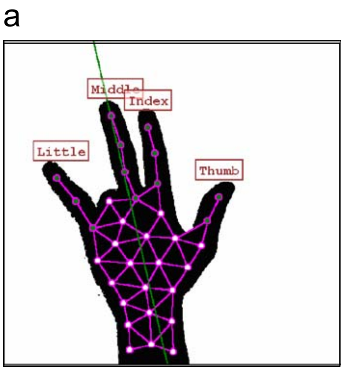
Fig. 20. (a) Recognition of a gesture with two raised fingers and (b) possibilities of every one feasible gesture.



b

Little	Ring	Middle	Index	Thumb	Score
-	-	x	x	x	74.818
-	x	-	x	x	60.512
x	-	-	x	x	52.399
-	x	x	-	x	46.274
x	-	x	-	x	38.161
x	x	-	-	-	33.26
-	x	x	x	-	13.269
x	-	x	x	-	5.157
x	x	-	x	-	0.255
x	x	x	-	-	0.003

Fig. 21. (a) Recognition of a gesture with three raised fingers and (b) possibilities of every one feasible gesture.



b

Little	Ring	Middle	Index	Thumb	Score
x	-	x	x	x	99.726
x	x	-	x	x	74.636
-	x	x	x	x	67.174
x	x	x	-	x	66.671
x	x	x	x	-	43.602

Fig. 22. (a) Recognition of a gesture with four raised fingers and (b) possibilities of every one feasible gesture.

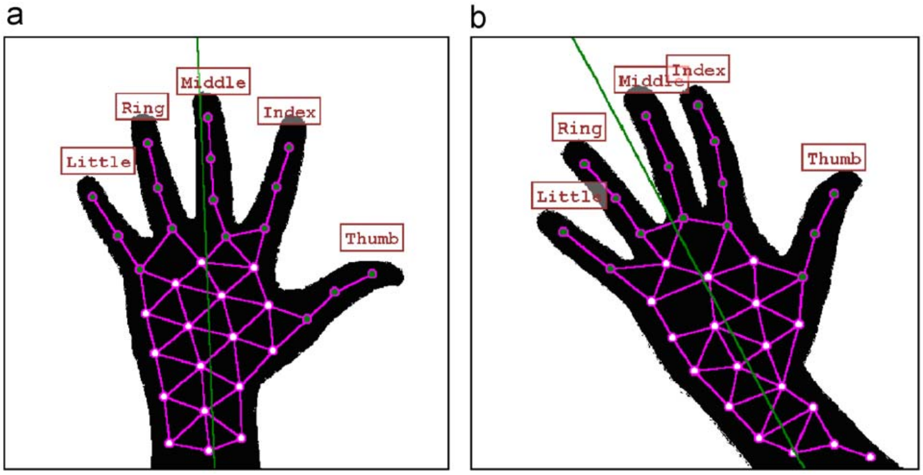


Fig. 23. (a) and (b) Recognition of gestures with five raised fingers.

Hence an Index is often classified falsely as a Thumb or vice versa. It is worth, also, to mention that the RC angle has the highest finger recognition rate, because there is no significant overlapping of the probability distributions of the classes.

3.3. *Experiment 3*

Experiment 3 aims to determinate the proposed system's recognition rate. Therefore, the system is tested with 180 test hand images 1800 times. The recognition rate, under the conditions described in the beginning of Section 2, is 90.45%. This satisfactory recognition rate is due to the robustness of each one of the stages of the proposed method. The mistakes of the recognition process are due to false feature extraction and mainly due to false estimation of the hand slope.

Figs. 19–23 present a number of examples of the output images of the proposed gesture recognition system. It is obvious that the recognition is successful regardless of the slope of the hand. The average computation time required for recognition of a hand gesture is about 1.5 s, using a 3 GHz CPU.

4. Conclusions

This paper proposes a new technique for hand gesture recognition which is based on hand gesture features and on a neural network shape fitting procedure. Firstly, the hand region is isolated by using a skin color filtering procedure in the *YCbCr* color space. This is a very fast procedure that results in noiseless segmented images regardless to the variation of the skin color and the lighting conditions. The stage that concerns the fitting of the hand's shape as well as the stage of finger features extraction is based on the innovative and powerful Self-Growing and Self-Organized Neural Gas network which approximate the hand's morphology in a very satisfactory way. As a result, the extracted finger features are well discriminated, they are invariant to the hand's size and slope and thus they conduce to a successful recognition. Finally, the hand gesture recognition, which is based on the Gaussian distribution of the finger features, takes into consideration the possibility of a finger belonging to each one of the five feasible classes as well as the likelihood of all the feasible finger combinations to correspond to input hand gesture. It is found from the experiments that the recognition rate is very promising and approaches 90.45%.

As a result, it is worth to underline that the key characteristic of the proposed hand gesture recognition technique is the use of the SGONG neural network. The reason is twofold; SGONG is able to describe very effectively the shape of the hand, and thus allows the extraction of robust and effective features, and moreover it achieves it by converging faster than other networks.

References

Albiol, A., Torres, L., Delp, E., 2001. Optimum color spaces for skin detection. In: IEEE International Conference on Image Processing, Thessaloniki, Greece, pp. 122–124.

Atsalakis, A., Papamarkos, N., 2005a. Color reduction by using a new self-growing and self-organized neural network. In: VVG05: Second International Conference on Vision, Video and Graphics, Edinburgh, UK, pp. 53–60.

Atsalakis, A., Papamarkos, N., 2005b. Color reduction using a self-growing and self-organized neural gas. In: Ninth International Conference on Engineering Applications of Neural Networks, Lille, France, pp. 45–52.

Atsalakis, A., Papamarkos, N., Andreadis, I., 2005. Image dominant colors estimation and color reduction via a new self-growing and self-organized neural gas. In: CIARP: Tenth Iberoamerican Congress on Pattern Recognition, Havana, Cuba, pp. 977–988.

Atsalakis, A., Papamarkos, N., 2006. Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural GAS. Engineering Applications of Artificial Intelligence 19, 769–786.

Chai, D., Ngan, K.N., 1998. Locating facial region of a head-and-shoulders color image. In: Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, pp. 124–129.

Chai, D., Ngan, K.N., 1999. Face segmentation using skin color map in videophone applications. IEEE Transactions on Circuits and Systems for Video Technology 9, 551–564.

Chen, F.S., Fu, C.M., Huang, C.L., 2003. Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Vision Computer 21 (8), 745–758.

Doulamis, N., Doulamis, A., Kosmopoulos, D., 2005. Content-based decomposition of gesture videos. In: IEEE Workshop on Signal Processing Systems, Athens, Greece, pp. 319–324.

Fritzke, B., 1994. Growing cell structures—a self-organizing network for unsupervised and supervised learning. Neural Networks 7 (9), 1441–1460.

Fritzke, B., 1995. A growing neural gas network learns topologies. In: Tesauro, G., Touretzky, D. S., Leen, T. K. (Eds.), Advances in Neural Information Processing Systems, vol. 7, MIT Press, Cambridge, UK, pp. 625–632.

Herpers, R., Derpanis, K., MacLean, W.J., Verghese, G., Jenkin, M., Milios, E., Jepson, A., Tsotsos, J.K., 2001. SAVI: an actively controlled teleconferencing system. Image and Vision Computing (19), 793–804.

Hongo, H., Ohya, M., Yasumoto, M., Yamamoto, K., 2000. Face and hand gesture recognition for human–computer interaction. In: ICPR00: Fifteenth International Conference on Pattern Recognition, Barcelona, Spain, pp. 2921–2924.

Huang, C.H., Huang, W.Y., 1998. Sign language recognition using model-based tracking and a 3D Hopfield neural network. Machine Vision and Applications (10), 292–307.

Huang, C.L., Jeng, S.H., 2001. A model-based hand gesture recognition system. Machine Vision and Applications (12), 243–258.

Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J., 1992. Comparing images using the Hausdroff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 437–452.

Kjeldsen, R., Kender, J., 1996. Finding skin in colour images. In: IEEE Second International Conference on Automated Face and Gesture Recognition, Killington, VT, USA, pp. 184–188.

Kohonen, T., 1990. The self-organizing map. Proceedings of IEEE 78 (9), 1464–1480.

Kohonen, T., 1997. Self-Organizing Maps, second ed. Springer, Berlin.

Licsar, A., Sziranyi, T., 2005. User-adaptive hand gesture recognition system with interactive training. Image and Vision Computing 23 (12), 1102–1114.

Manresa, C., Varona, J., Mas, R., Perales, F.J., 2000. Real-time hand tracking and gesture recognition for human–computer interaction. Electronic Letters on Computer Vision and Image Analysis (0), 1–7.

O' Mara, D.T.J., 2002. Automated facial metrology. Ph.D. Thesis, Department of Computer Science and Software Engineering, University of Western Australia.

Stergiopoulou E., Papamarkos, N., 2006. A new technique for hand gesture recognition. In: ICIP2006: International Conference on Image Processing, Atlanta, USA.

Tan, R., Davis, J.W., 2004. Differential video coding of face and gesture events in presentation videos. Computer Vision and Image Understanding 96 (2), 200–215.

Triesch, J., Von der Malsburg, C., 2001. A system for person-independent hand posture recognition against complex backgrounds. IEEE Transanctions on Pattern Analysis and Machine Intelligence 23 (12), 1449–1453.

Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M., Handler, J., 2005. A real-time hand gesture system based on evolutionary search. In: GECCO2005: Tenth Genetic and Evolutionary Computation Conference, Washington, DC, USA.

Xiaoming, Y., Ming, X., 2003. Estimation of the fundamental matrix from uncalibrated stereo hand images for 3D hand gesture recognition. Pattern Recognition 36, 567–584.

Yoon, H.S., Soh, J., Bae, Y.J., Yang, H.S., 2001. Hand gesture recognition using combined features of location, angle and velocity. Pattern Recognition 34 (7), 1491–1501.

Yoruk, E., Konukoglu, E., Sankur, B., Darbon, J., 2006. Shape-based hand recognition. IEEE Transactions on Image Processing 15 (7), 1803–1815.