

## COLOR FACE SEGMENTATION USING A FUZZY MIN-MAX NEURAL NETWORK

JUAN WACHS\* and HELMAN STERN†

*Department of Industrial Engineering and Management,  
Ben-Gurion University of the Negev, Be'er-Sheeva, Israel*

*\*juan@bgumail.bgu.ac.il*

*†helman@bgumail.bgu.ac.il*

MARK LAST

*Department of Information Systems Engineering,  
Ben-Gurion University of the Negev, Be'er-Sheeva, Israel*

*mlast@bgumail.bgu.ac.il*

Received 17 June 2002

Revised 8 September 2002

This work presents an automated method of segmentation of faces in color images with complex backgrounds. Segmentation of the face from the background in an image is performed by using face color feature information. Skin regions are determined by sampling the skin colors of the face in a Hue Saturation Value (HSV) color model, and then training a fuzzy min-max neural network (FMMNN) to automatically segment these skin colors. This work appears to be the first application of Simpson's FMMNN algorithm to the problem of face segmentation. Results on several test cases showed recognition rates of both face and background pixels to be above 93%, except for the case of a small face embedded in a large background. Suggestions for dealing with this difficult case are proffered. The image pixel classifier is linear of order  $O(Nh)$  where  $N$  is the number of pixels in the image and  $h$  is the number of fuzzy hyperbox sets determined by training the FMMNN.

*Keywords:* Skin segmentation; color recognition; face detection; fuzzy logic; fuzzy clustering; pattern recognition; fuzzy neural networks.

### 1. Introduction

Skin color segmentation for head or hand tracking is a complex problem since it includes several factors like non-uniform (complex) backgrounds, object movement, illumination of the scene, and noise at high frequency sampling. In complex backgrounds there frequently exist objects with similar color distributions (e.g. the skin color) and therefore the results obtained can include foreign objects that the user did not mean to track. Some methods based on the topological head structure can discriminate these foreign objects, but even this topological head structure varies from front and side views. The direction, the angle, and the velocity of the movement

requires a different methodology for systems working in real time. Illumination of the scene has an important influence in the color space model used, especially the RGB model. The goal of this work is to develop a skin color segmentation algorithm capable of detecting faces at full-frame size and rate relying on the HSV color space model and the Fuzzy Min Max Neural Network algorithm. This work can be further implemented for tracking human hands and faces.

In this paper, we present an automated method of segmentation of faces in color images with complex backgrounds. Segmentation of the face from the background in an image is performed by using face color feature information. Skin regions are determined by sampling the skin colors of the face in a Hue Saturation Value (HSV) color model, and then training a fuzzy min-max neural network (FMMNN) to automatically segment these skin colors. This work appears to be the first time that Simpson's FMMNN algorithm<sup>6-8</sup> has been applied to the problem of face segmentation. Some problems that are relevant to this task are: finding a fast classification algorithm, high performance in complex backgrounds, high performance for non fixed head sizes (including zoom in and out of the face) and robustness to different toned skin faces (dark and white skin color).

The remainder of the paper is organized as follows. Section 2 provides a survey of related work. Section 3 presents the methodology used in terms of a color space model and color distribution model learning using a fuzzy min-max clustering neural network (FMMNN). Section 4 describes the system performance measure for testing the FMMNN. Results of a three case experiment are provided in Sec. 5. Finally, the paper's conclusions are given in Sec. 6.

## **2. Related Work**

For segmentation of skin-like regions it is possible to use appropriately defined domains of hue and saturation. By disregarding the value (luminance) component, robustness is obtained with respect to changes in illumination and shadows. An example for such segmentation is found in Ref. 1. They assume that the face is a connected region with skin color, and therefore connected component analysis on the segmented image is performed. Real time results were obtained by tracking faces over 150 frames, with a rate of 1/10 frames per second. Their system is based on the assumption that the HS skin color distribution is continuous over the spectrum, and therefore, foreign objects also are recognized as faces. The Hue Saturation Value (HSV) color model was also used in Ref. 2 where it takes the form of a statistical distribution property of skin colors. For face tracking via a flesh color model, flesh areas from the user are sampled by prompting users to center their face in an on-screen box, or by using motion cues to find flesh areas from which to sample colors. The hues derived from flesh pixels in the image are sampled from the H channel and binned into a 1D histogram. When sampling is complete, the histogram is saved for future use. During operation, the stored flesh color histogram is used as a model,

or lookup table, to convert incoming video pixels to a corresponding probability of being a flesh image. Using this method, probabilities range in discrete steps from zero (probability 0.0) to the maximum probability pixel value (probability 1.0). For 8-bit hues, this range is between 0 and 255. Then they track using the CAMSHIFT algorithm on the probability of a flesh image. The main deficiency of this system is that a tedious setup process must be performed for every new user, who has to center his/her face in a window to prepare the distribution histogram model.

A similar approach of this lookup table can be found in Ref. 3. A universal skin-color map is derived and used on the chrominance component of the input image to detect pixels with skin-color appearance. Based on the spatial distribution of the detected skin-color pixels and their corresponding luminance values, the algorithm employs a set of regularization processes to reinforce regions of skin-color pixels that are more likely to belong to the facial regions and eliminate those that are not. This work is unable to deal with complex backgrounds, since it relies on the fact that the background region tends to a uniform distribution of brightness.

Classification methods were found to be very useful under the assumption that skin color has a distribution of small regions that form clusters at specific points over the HSV three dimensional plane. Different classification methods look for these clusters to form “classes” to which every pixel belongs. Neural networks (NN) are a classical method for data classification for skin color segmentation as shown in Ref. 4. The RCE neural network adaptive pattern classification engine which is composed of three cells on the input layer of the network is designed to represent the  $L * a * b^*$  color values of a pixel in the image. The middle layer cells are called prototype cells, and each cell contains color information about an example skin color class that occurred in the training data. The cell on the output layer corresponds to the skin color class. During the training procedure, the RCE network allocates the positions of prototype cells and modifies the sizes of their corresponding spherical influence fields, so as to cover the arbitrarily complex distribution region of skin colors in the color space. During running, RCE responds to input color signals in the fast response mode. If an input color signal falls into the distribution region of skin colors, this input color signal belongs to the skin color class, and the pixel represented by this color signal is identified as skin color in the image. One important drawback of this system is the time cost of the algorithm, for a high-resolution image it takes close to one second to segment the skin color, which is too long for a real-time system. Also Ref. 5 is based on NN. A network was trained to detect frontal or near-frontal views. Other networks could be similarly trained to detect faces at different pose angles in order to build a hierarchical view-based detector exhibiting pose invariance. Face training images were normalized with respect to orientation and scale. A total of 9000 face images were used during training. A back-propagation NN algorithm, with iterative selection of false-positive non-faces was trained to detect frontal or near-frontal views. One of the weaknesses inherent in this form of static scene detector is that there exist non-face images patches which when taken out of their spatial and temporal context appear “face-like”.

Cluster classification algorithms can be used in data partitions of photometric representations, which model skin color pixels as points in large multi-dimensional hyperspaces. Since the skin color values vary within the face, distribution data could be clustered to more than one cluster, therefore the Fuzzy Min-Max Neural Network (FMMNN)<sup>6–8</sup> is a good candidate for the skin color segmentation task. Also the FMMNN showed short classification times when used in real-time systems.<sup>9,10</sup> The FMMNN was used in these systems for hand gesture classification where posture data are very sensitive to the hand size of the signer. To overcome these variations, online adaptation capability is required for the posture classifier, making NN and supervised clustering algorithms obsolete.

### 3. Methodology

#### 3.1. General

The system methodology is to capture human faces in a laboratory via a standard PAL camera. The camera signal is then digitized in RGB and transformed to the Hue Saturation Value (HSV) color space. HSV space separates out hue (color) from saturation (how concentrated the color is) and brightness. The data patterns for future use in the classifier uses only the  $H$  (hue) and  $S$  (saturation) channels of the HSV space. This eliminates the  $V$  (value) channel, which is only influenced by illumination changes, and hence can provide a more robust recognition system.

A training set of the HS pair face skin color data is obtained using a sample image of a face. This face image is extracted using a mask, from a more complex sample image containing a face and a non-uniform background. In the training stage, the fuzzy min-max clustering neural network converges into pattern clusters after only a few passes through the data set (Fig. 1). All the clusters created in the training stage, belong to one class, the face skin color class. Once the clusters are created, the recognition stage can commence. Face pixels are segmented via the Fuzzy Min-Max Neural Network clustering algorithm (FMMNN).<sup>6,7</sup> As in the

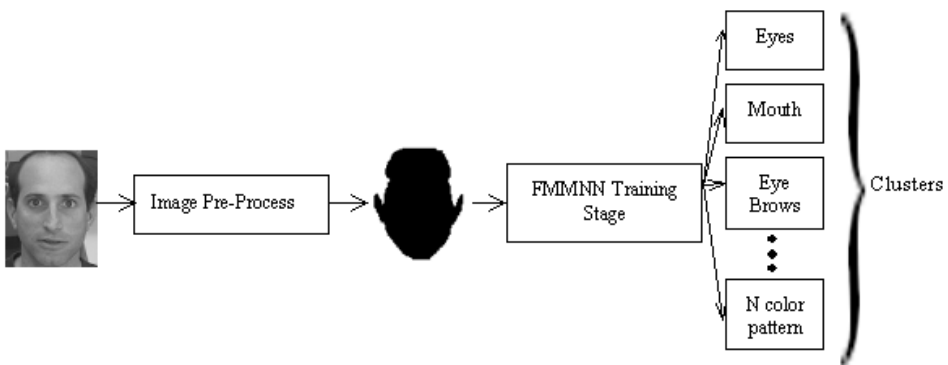


Fig. 1. Training Stage.

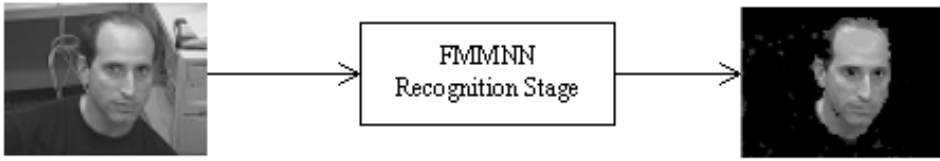


Fig. 2. Recognition stage.

training stage, the HS channels are kept while the  $V$  channel is discarded. Each HS pair from the data set is supplied to the FMMNN module, which returns a membership function. Any value in the membership function higher than some threshold will determine that the pixel (HS pair) belongs to a skin face region. In this way a segmentation process is performed (Fig. 2).

### 3.2. The fuzzy min-max neural network algorithm

Since the fuzzy min-max clustering neural network (FMMNN) is not well known, we present its explanation following Simpson.<sup>6,7</sup> The FMMNN uses hyper box fuzzy sets. A hyper box defines a region of the  $n$ -dimensional pattern space and all patterns contained inside the hyper box have full cluster membership. A hyper box is defined by its min point and max point. The combination of the min-max points and the hyper box membership function defines fuzzy sets (fuzzy clusters). The FMMNN can be viewed as a two layer neural network where the first layer contains  $n$  input elements (one for each dimension of the input pattern being clustered). The output layer contains  $m$  processing elements (one for each cluster). Two links connect each input to each output. The weights on the links are the min and max vectors (corners) of the hyperbox represented by the output elements. The output elements correspond to fuzzy set hyperboxes. The output element values represent the degree to which the input pattern  $A_k = (a_{k1}, a_{k2}, \dots, a_{kn})$  belongs to each of the  $m$  clusters. Although Simpson gives this model a neural network interpretation, the classical neural network weight learning algorithms using a “supervised” training set of input-output patterns is not used. Instead, an unsupervised method of determining the number and properties of the output elements is used and is described below.

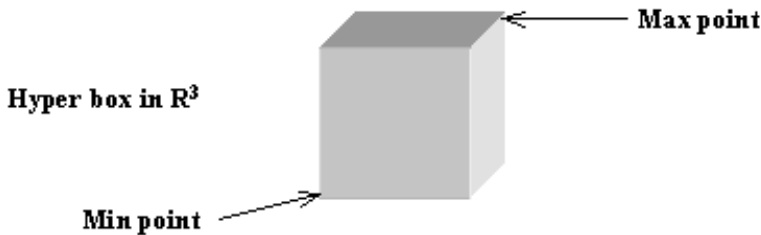


Fig. 3. The min-max hyper box for  $R^3$ .

Learning in the FMMNN consists of creating and adjusting hyper boxes in pattern space as they are received. Once the FMMNN has learned the clusters through unsupervised training, it is operated by presenting an input pattern and computing the pattern’s membership grade in each of the existing fuzzy hyper box sets. An illustration of a three-dimensional hyper box is shown in Fig. 3.

Let the  $j$ th hyper box fuzzy set,  $B_j$ , be defined by the ordered set:

$$B_j = \{A_k, V_j, W_j, b_j(A_k, V_j, W_j)\}. \tag{1}$$

Let there be  $m$  patterns indexed as  $k = 1, 2, \dots, m$ , Let  $A_k = (a_{k1}, a_{k2}, \dots, a_{kn})$  denote the  $k$ th pattern in the data set,  $V_j = (v_{j1}, v_{j2}, \dots, v_{jn})$  and  $W_j = (w_{j1}, w_{j2}, \dots, w_{jn})$  denote the min and max points of the  $j$ th hyper box, respectively. The membership function for the  $j$ th hyper box is

$$0 \leq b_j(A_k, V_j, W_j) \leq 1. \tag{2}$$

The membership function measures the degree to which the  $k$ th input pattern  $A_k$  falls within the hyper box formed by the min point  $V_j$  and the max point  $W_j$ . As  $A_k$  approaches the hyper box,  $b_j(A_k, V_j, W_j)$  approaches 1, and when the point is contained within the hyper box,  $b_j(A_k, V_j, W_j) = 1$ . The resulting membership function is given by Eqs. (3) and (4). The parameter  $\gamma$  is the sensitivity parameter, representing the fuzziness of the membership function and it affects the steepness of the ramp approaching the sides of the hyperbox:

$$b_j(A_k, V_j, W_j) = \frac{1}{n} \sum_{i=1}^n (1 - f(a_{ki} - w_{ji}, \gamma) - f(v_{ji} - a_{ki}, \gamma)) \tag{3}$$

$$f(x, y) = \begin{cases} 1 & \text{if } xy > 1, \\ xy & \text{if } 0 \leq xy \leq 1, \\ 0 & \text{if } xy < 1. \end{cases} \tag{4}$$

Learning in the fuzzy min-max clustering neural network is an expansion/contraction process. Assume that the training set  $\mathbf{A}$  consists of  $m$  patterns, where  $A_k = (a_{k1}, a_{k2}, \dots, a_{kn})$  is the  $k$ th pattern. The learning process begins by selecting an exemplar for  $\mathbf{A}$  and finding the closest hyper box to that pattern, that can expand (if necessary) to include the pattern. If a hyper box that meets the expansion criteria cannot be found, a new hyper box is formed and added to the system. The growth process allows existing clusters to be refined over time, and it also allows new clusters to be added without retraining. In case the two hyper boxes overlap each other, the contraction process is started, since overlap causes ambiguity.

### 3.2.1. Fuzzy min-max learning

The learning algorithm includes the following stages:

*Initialization:* Initialize all the min-max points prior to any learning.

*Expansion:* Identify the hyper boxes closest to the input pattern that can, within constraints, be expanded and expand them. If an expandable hyper box cannot be found, add a new hyper box.

*Overlap test:* Determine whether the recent expansion caused any overlap between hyper boxes.

*Contraction:* If the expansion test identified any overlapping hyper boxes, contract the hyper boxes to eliminate the overlap.

A brief description of each stage follows.

### 3.2.2. Hyper box initialization

The min points are initially  $V_j = 1$  and the max points initially are  $W_j = 0$  for all  $B_j$ . In the first iteration we will obtain  $V_j = W_j = A_k$ . Thus, each input pattern is initially defined as a hyper box.

### 3.2.3. Hyper box expansion

For a hyper box  $B_j$  to expand to include  $A_k$ , the constraint defined in Eq. (5) must be met:

$$\sum_{i=1}^n (\max(w_{ji}, a_{ki}) - \min(v_{ji}, a_{ki})) \leq n\theta. \tag{5}$$

The user defined value,  $\theta$  lies in the interval  $[0, 1]$  and represents the maximum size of a hyperbox.

If all  $B_j$  are exhausted without any possible expansions, then create a new hyper box  $B_{j+1}$ , otherwise expand the hyper box to the new min-max points described in Eq. (6):

$$\begin{aligned} v_{ji}^{\text{new}} &= \min(v_{ji}^{\text{old}}, a_{ki}) & \forall i = 1, 2, \dots, n, \\ w_{ji}^{\text{new}} &= \max(w_{ji}^{\text{old}}, a_{ki}) & \forall i = 1, 2, \dots, n. \end{aligned} \tag{6}$$

### 3.2.4. Hyper box overlap test

Assume the hyper box  $B_j$  is expanded in the previous step. To determine overlap, a dimension-by-dimension comparison is conducted between  $B_j$  and the remaining  $B_k$  to check whether one of the four test cases is satisfied for each of the  $n$  dimensions:

- Case a:  $v_{ji} < v_{ki} < w_{ji} < w_{ki}$ ,
- Case b:  $v_{ki} < v_{ji} < w_{ki} < w_{ji}$ ,
- Case c:  $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$ ,
- Case d:  $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$ .

### 3.2.5. Hyper box contraction

Case a:  $v_{ki}^{\text{new}} = w_{ji}^{\text{new}} = (v_{ki}^{\text{old}} + w_{ji}^{\text{old}})/2$ ,

Case b:  $v_{ji}^{\text{new}} = w_{ki}^{\text{new}} = (v_{ji}^{\text{old}} + w_{ki}^{\text{old}})/2$ ,

Case c: If  $w_{ki} - v_{ji} < w_{ji} - v_{ki}$ , then  $v_{ji}^{\text{new}} = w_{ki}^{\text{old}}$ , otherwise  $w_{ji}^{\text{new}} = v_{ki}^{\text{old}}$ ,

Case d: If  $w_{ji} - v_{ki} < w_{ki} - v_{ji}$ , then  $v_{ki}^{\text{new}} = w_{ji}^{\text{old}}$ , otherwise  $w_{ki}^{\text{new}} = v_{ji}^{\text{old}}$ .

Cluster convergence occurs when all hyper box *min* and *max* points remain unchanged during successive presentations of the data set in the same order.

## 4. System Evaluation

This part consists of two fundamental stages — training and testing of the FMMNN. A performance measure is also proffered for system evaluation. The FMMNN is trained using a sample image containing a bright-skin face. The system will be tested with three images. In the first case the test image used is identical to the face used in the training stage, but with a different posture. In the second case the face presented is different from the one used in training. The third case has a face image embedded in a large background. The classification rule is applied to each pixel in the image. The classification rule is: If the pixel belongs to some hyperbox with a membership grade higher than some threshold  $\tau$ , then the pixel is classified as belonging to the “skin face color” category, otherwise it is classified as a background pixel. For  $h$  hyperboxes and a pixel pattern  $k$  the rule is

*If  $\text{Max}\{b_j(A_k, V_j, W_j), j = 1, \dots, h\} > \tau$  then the pixel  $k$  is a face pixel.*

If there are  $N$  pixels in the image, the complexity of the classifier is of the order  $O(Nh)$ . The performance is defined as a recognition rate using the number of the skin-color pixels recognized plus the background color pixels recognized over the total number of image pixels as shown by

*recognition rate*

$$= \left( 1 - \frac{\text{unrecognized face pixels}}{\text{total pixels}} - \frac{\text{unrecognized background pixels}}{\text{total pixels}} \right) \times 100\% \quad (7)$$

## 5. Results

The system was trained using the HS matrix of the face skin portion of an image as the training set (Fig. 4). This portion was extracted using a mask (Fig. 5). The training image consists of 8528 pixels, of which 5027 pixels belong to the human face (59% of the image). The image is downsized to 520 pixels in order to reduce computation time. Therefore, the input matrix of the classifier has 304 HS pair vectors (59% of 520 pixels).



Fig. 4. Face image.

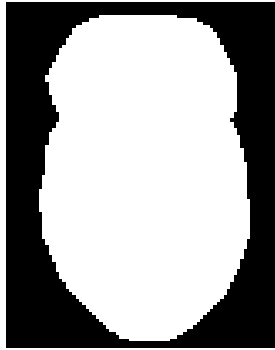


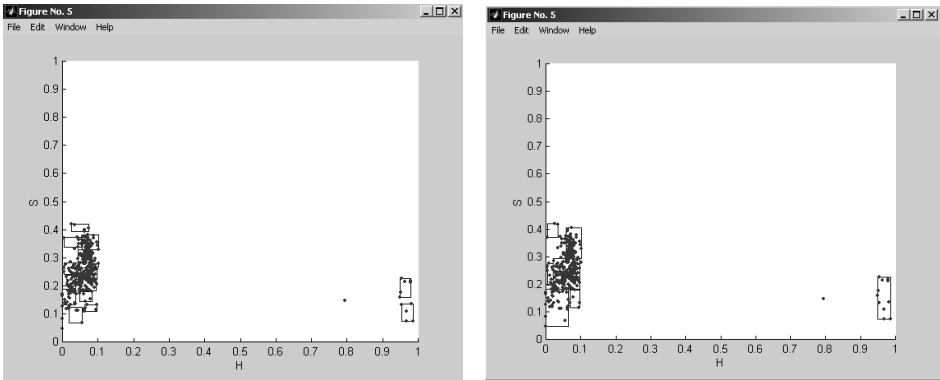
Fig. 5. Skin color region.

For the face part of the training image, it is possible to see the hyperbox distribution in HS space. For  $\theta$  equal to 0.05, 19 hyperboxes were created, and for  $\theta$  equal to 0.1, 8 hyperboxes were created (see Fig. 6). In Fig. 6, although the set of hyperboxes appears disjoint, they actually form a compact cluster of colors because of the wrap-around property of the hue dimension. Note, that since a larger  $\theta$  allows larger hyperboxes to be formed there are fewer of them.

After the training stage the system was tested using three different images. The best value of the  $\theta$  parameter was found empirically to be 0.01. The threshold is a variable parameter that controls the number of clusters generated.

**Case 1.** An input image of the same face used in training but with a three-quarter profile viewed slightly from above. This image consists of 4,466 pixels (see Fig. 7).

For each pixel of this image, in the HS space color dimension, the membership function of each hyperbox is evaluated. A new image is created including all the pixels of the tested image, where the skin color pixels are left with the **original color**, while the others (those with low degree of membership values) left as **black color** pixels. Figure 8 shows the segmentation of the test image (Fig. 9)



(a)  $\theta = 0.05$ .

(b)  $\theta = 0.1$ .

Fig. 6. Hyperbox distributions.



Fig. 7. First test image.



Fig. 8. First segmented image.

For this case, the following parameters were used:  $\theta = 0.01$ ,  $\tau = 0.95$ . The recognition accuracy obtained using Eq. (7) is

$$\text{recognition rate} = \left( 1 - \frac{21}{4466} - \frac{289}{4466} \right) \times 100\% = 93.06\%.$$



Fig. 9. Second test image.



Fig. 10. Second segmented image.

**Case 2.** The second testing image corresponds to a dark skin person. The process is the same as explained earlier. This image consists of 6,912 pixels (see Fig. 9).

Figure 10 shows the results of the segmentation of the image using  $\theta = 0.01$ ,  $\tau = 0.95$ .

The recognition accuracy obtained using Eq. (7) is

$$\text{recognition rate} = \left( 1 - \frac{40}{6912} - \frac{171}{6912} \right) \times 100\% = 96.95\%.$$

**Case 3.** The third test image includes a distant face with a large background in a laboratory. This image consists of 27,648 pixels (see Fig. 11).

Figure 12 shows the results of the segmentation of the image using  $\theta = 0.01$ ,  $\tau = 0.99$ .

The recognition accuracy obtained using Eq. (7) is

$$\text{recognition rate} = \left( 1 - \frac{80}{27648} - \frac{2778}{27648} \right) \times 100\% = 89.66\%.$$

## 6. Conclusions

This work presents an application of the FMMNN algorithm to image face segmentation by color. This approach is based on the face skin color uniform distribution



Fig. 11. Third test image.



Fig. 12. Third segmented image.

property exposed using the HSV color space model. The system was trained on a face image of a bright-skin person. Three cases were examined to evaluate the performance of the system: First, an image where the face of the same person is the main blob object. In this case, the system has reached a recognition accuracy of about 93%. Second, for the image where the main object is a colored face of a dark-skin person the result of 96% was even better. For the third case (the bright-skin face in a large background), the results were quite disappointing with a recognition rate of about 89%. The best value of the theta parameter was found empirically to be 0.01. The classifier accepts a user defined fuzzy membership level threshold for determining if a pixel is classified as a face or background pixel. This classifier is linear of order  $O(Nh)$  where  $N$  is the number of pixels in the classified image and  $h$  is the number of fuzzy hyperboxes.

The trained network appears to perform well for those images, where the face, like in the training image, is the main object. However, for those images where

the object is far from the camera, the system's performance drops quickly. To recover the lost accuracy, one may try to train the system on a larger and more diverse set of face images than the one used in our experiments. However, a more effective approach for images with large backgrounds is to apply spatial clustering and remove pixels far from the cluster's centroid. Eliminating shadows and black pixels from the training face which may also appear in the background may help alleviate this problem. If the background has a lot of pixels similar to face skin colors, then other cues must be used such as motion, local texture, shape and relative position (if prior knowledge is available). This presents no problem to our classifier, as only the dimension of the pattern and the resultant hyperboxes need be increased to accommodate the increased pattern size.

A problematic issue of the system is the dynamic cluster creation property of the FMMNN algorithm. Since the number of clusters is not defined a priori, the algorithm creates clusters whenever needed. This is an advantage because different color details of the face can be presented, without a limit. On the other hand, this is an important limitation, since rare colors will have the same representation as common ones, and therefore, it is possible to find these rare colors in the background and treat them as face pixels.

It is possible to resolve this problem in order to improve the performance significantly. One way is to add a new parameter in each hyperbox representation that expresses the compactness of the hyperbox, and the size of its population. A more compact and highly populated hyperbox is more likely to represent a skin color cluster. When each pixel is classified as belonging to the face or not, in addition to the membership grade of that pixel, the value expressing compactness and population size of the corresponding cluster can also be considered.

## Acknowledgments

This project was partially supported by the Paul Ivanier Center for Robotics Research and Production Management, Ben-Gurion University of the Negev.

## References

1. J. Sobottka and I. Pittas, "Segmentation and tracking of faces in color images," in *Int. Conf. Automatic Face and Gesture Recognition* (1996), pp. 235–241.
2. G. Bradski, *Computer vision face tracking for use in a perceptual user interface* (Intel Technology Journal, 2nd Quarter, 1998).
3. D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. Circuits Syst. Video Tech.* **9**(4), 551–564 (June 1999).
4. X. M. Yin and M. Xie, "Finger identification in hand gesture based human-robot interaction," *J. Robotics and Autonomous Systems* **34**(4), 235–250 (March 2001).
5. S. McKenna and S. Gong, "Tracking faces," in *Int. Conf. Automatic Face and Gesture Recognition* (October 1996), pp. 271–276.
6. P. K. Simpson, "Fuzzy min-max neural networks — part I: Classification," *IEEE Trans. Neural Networks* **3**, 776–786 (1992).

7. P. K. Simpson, "Fuzzy min-max neural networks — part 2: Clustering," *IEEE Trans. Fuzzy Systems* 1(1), 32–45 (1993).
8. R. C. Eberhart, R. W. Dobbins, and P. K. Simpson, *Computational Intelligence PC Tools* (Academic Press Professional, 1996), pp. 308–318.
9. C.-S. Lee, G.-T. Park, J.-S. Kim, Z. Bien, W. Jang, and S.-K. Kim, "Real-time recognition system of korean sign language based on elementary components," in *Proc. Sixth Int. Conf. Fuzzy Systems* (IEEE FUZZ'97, Spain, 1997), pp. 1463–1468.
10. J.-S. Kim, K.-H. Park, J.-B. Kim, J.-H. Do, K.-J. Song, and Z. Bien, "Study on intelligent autonomous navigation of Avatar using hand gesture recognition," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics* (Nashville, USA, 2000), pp. 846–851.



**Helman Stern** received a Ph.D. in Operations Research from the University of California, Berkeley (1969), a B.S. in Electrical Engineering from Drexel University (1957), and a Masters in Engineering Administration from George Washington University (1961). He has taught at Rensselaer Polytechnic Institute, the University of California, Berkeley, SUNY at Albany and San Francisco State University. He has been teaching at Ben Gurion University of the Negev since 1974. He is a member of, ACM, SigMultimedia, INFORMS and IEEE Computer Society. He has publications in such journals as Pattern Recognition Letters, Networks, Computers and OR, Management Science, Operations Research, The Journal of Optimization Theory and Application, Human Factors, and Imaging Science. His current research interests are in image processing, machine vision, intelligent multimedia systems, computer vision, hand gesture recognition, image and video databases, face tracking, tele-operations, and scheduling. At BGU he is head of the Intelligent Systems Program and teaches the courses, Intelligent Systems, and Machine Vision.



**Mark Last** received his M.Sc. (1990) and Ph.D. (2000) degrees in Industrial Engineering from Tel Aviv University, Israel. He is currently a Lecturer at the Department of Information Systems Engineering, Ben-Gurion University of the Negev, Israel. Prior to that, he was a visiting Assistant Professor at the Department of Computer Science and Engineering, University of South Florida, USA (1999–2001), a Senior Consultant in Industrial Engineering and Computing (1994–1998), and the Head of Production Control Department at AVX Israel (1989–1994). Mark Last has published over 40 papers and chapters in journals, books, and conferences. He is a co-author of the book "Knowledge Discovery and Data Mining — The Info-Fuzzy Network (IFN) Methodology" (Kluwer, 2000) and a co-editor of the book "Data Mining and Computational Intelligence" (Physica-Verlag, 2001). His current research interests include data mining, fuzzy logic, and software testing.



**Juan Wachs** received a B.Ed.Tech (1995) and is currently a Ph.D. candidate in Intelligent Systems in the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel where he is a teaching assistant in Machine Vision and CIM Automation courses. He has become a critical member of a team of researchers working on Virtual Reality Telerobotic Operations through the use of a hand gesture-computer interface. Based on his thesis, Juan Wach's paper "Real-Time Hand Gesture Telerobotic System Using the Fuzzy C-Means Clustering Algorithm" (with U. Kartoun, H. Stern, and Y. Edan) received the best student award at the 12th Annual Conference of Industrial Engineering and Management, Tel-Aviv, Israel. His current research interests include machine-vision, telerobotics and virtual reality.