Recurrent Neural Network Verifier for Face Detection and Tracking

Sung H. Yoon[1], Gi T. Hur[2], and Jung H. Kim[3]

[1] Computer Science Dept., North Carolina A&T State University, NC 27411, USA
[2] Multimedia Contents Dept., DongShin University, South Korea
[3] Electrical Engineering Dept., North Carolina A&T State University, NC 27411, USA

Abstract. This paper presents a new method for face verification for vision applications. There are many approaches to detect and track a face in a sequence of images; however, the high computations of image algorithms, as well as, face detection and head tracking failures under unrestricted environments remain to be a difficult problem. We present a robust algorithm that improves face detection and tracking in video sequences by using geometrical facial information and a recurrent neural network verifier. Two types of neural networks are proposed for face detection verification. A new method, a three-face reference model (TFRM), and its advantages, such as, allowing for a better match for face verification, will be discussed in this paper.

Keywords: face tracking; face verification; Hopfield neural networks; annealing; geometrical feature, video sequence.

1 Introduction

Pattern recognition and computer vision theory have been considerably improved during the last decade, such that, the appearance of an automated vision system seems very close to our future. However, because of the higher computational burden of understanding image algorithms, applications of face recognition to multi-context images are still limited to the restricted environment. Face detection and recognition have been an important issue in video sequence applications because different facial views and varied illumination cause problems when detecting and recognizing a human face. There are many approaches to detect and verify a human face in video sequences [1]-[8]. Some methods are based on feature invariants, which are used to find out structural features. Some are based on template matching, which uses a stored pattern to track head positions. Others include an appearance-based method that utilizes a trained model from a set of images to capture the representative variability of a facial appearance. The face detection and tracking approach developed in this study is based on using feature invariants: color, gradient and geometrical facial features[9].

We use geometric feature points for face verification based on neural networks. The feature points are usually detected in a curvature function space by capturing all the local extrema whose curvature values are above a certain threshold value. Geometric feature points can generate useful local features such as angles between neighboring corners and distances between every pair of corners. In addition, the geometric feature

M. Ali and R. Dapoigny (Eds.): IEA/AIE 2006, LNAI 4031, pp. 488 – 499, 2006.

points are appropriate for constructing neural networks. To extract reliable features, it is very important to detect consistent feature points invariant under translation, rotation, and scale. Our procedure for face detection finds candidates for face regions using color and gradient, and then extracts geometric features inside the face region. We model the face (mouth, nose, eyes) and use geometrical facial information for face verification. Geometrical facial feature properties change depending on image quality and size of a face.

The inherent parallelism of neural networks allows rapid pursuit of many hypotheses in parallel with high computation rate [10][11]. Moreover, it provides a great degree of robustness or fault tolerance compared to conventional computers because of many processing nodes, each of which is responsible for a small portion of the task. Damages to a few nodes or links thus do not impair overall performance significantly. For this reason, a Hopfield style neural network has been proposed to solve matching problems for face detection and recognition. It is one of the popular neural computations used in real world applications. Its popularity is due to its simple architecture and well-defined time-domain behavior. The Hopfield neural network (HNN) is composed of single-layer neurons with fully connected feedback connections. The neurons have the sigmoid gain characteristic, while the connectivity matrix corresponding to the connection is symmetric and the diagonal terms of the matrix are zero. Such networks always move in the direction of decreasing the energy of the networks and get stable states at the local minimum of energy.

In a complex background, there are many clutters. We use an innovative neural network based verifier for face detection. This technique can be further used for face recognition when it is required. In this paper, we propose a component-based approach to face detection and recognition in video sequences using a neural network verifier. Identifying a face with large tolerance proves the robustness of the algorithm. In Section 2, we describe the overall system description and the hybrid color scheme. In Section 3, we review the HNN, feature extraction and graph formation. In Section 4, we present our simulation results and the results are obtained. In Section 5, we state our conclusions.

2 Approach

2.1 Overall System Description

Under very restricted conditions, it is possible to track the human face with each modality alone. However, single modality results in substantial tracking failures in unconstrained environments. We have created a head tracking system that achieves robust performance via the integration of multiple visual features. By combining these modules in simple ways, we can build a system that attains overall robust performance [8]. We describe our system using three visual modalities: intensity gradient, skin color model, and geometrical face information.

In our experiment, tracking with color and gradient gives good results in restricted conditions [9]. However, it sometimes fails in unrestricted and complex environments. If we make use of a geometrical feature in addition to color and gradient, it can provide more robust head tracking results. Potential face regions detected from color and gradient are then verified by a geometrical feature. The geometrical feature of a face will enhance the detection rate of a face when it is used with color and gradient. It is sometimes difficult to accurately extract geometrical features from a face. We use a recurrent neural network for robust verification. Neural network based verification performs very well, even when some feature points are missing or off true positions. The following shows the overall description of the system. Figure 1 describes our algorithm to overcome the problem of frequent inaccurate predictions of a fast moving face's next position. The first step is to extract potential face regions using skin color and gradient. The next step is to extract facial features from the potential face regions. The final step is to verify if it is a true face region using the proposed neural network. A detailed description of each module is as follows:

We use the notation M for the current head's state and location. The following equation explains how to find the best head state and location within a search window.

$$M^* = \arg\max_{M_i}(\alpha(M_i) + \beta(M_i)) \tag{1}$$

where $\alpha(M_i)$, $\beta(M_i)$ are the matching scores based on intensity gradients and color histograms, respectively. The search space M is the set of all states within some range of the predicted location with simple linear prediction.

The candidates that have high matching scores are then verified by a neural network. This verification method is one of the major contributions of this research. This verification method is so robust that it could verify a face and non-face without accurate geometrical features. This verification technique can be further used for face recognition if the quality of an image is good enough to extract accurate feature points.

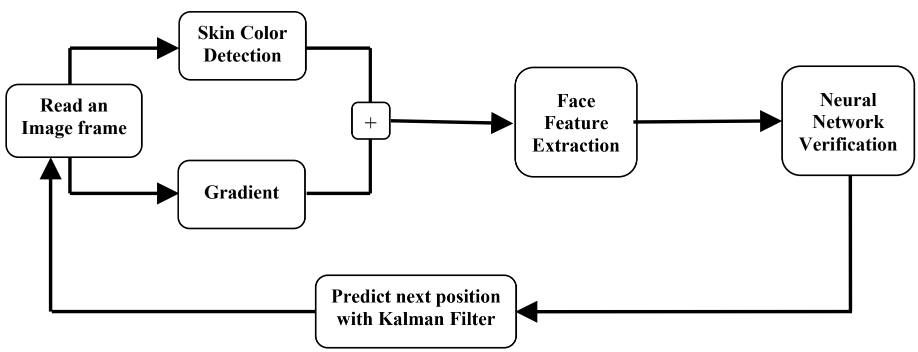


Fig. 1. Head tracking algorithm using image segmentation and Kalman Estimation

2.2 Neural Network Based Verification

The appearance of an automated vision system seems very close to our future because pattern recognition and computer vision theory has been considerably improved during the last decade. However, because of the higher computational burden of understanding image algorithms, applications of face recognition to multi-context images are still limited to the restricted environment. The Hopfield neural network is one of the popular neural computations used in real world applications. Its popularity is due to its simple architecture and well-defined time-domain behavior. The Hopfield neural network is composed of single-layer neurons with fully connected feedback connections. The neurons have the sigmoid gain characteristic, while the connectivity matrix corresponding to the connection is symmetric and the diagonal terms of the matrix are zero. Such networks always move in the direction of decreasing the energy of the networks and get stable states at the local minimum of the energy. The face verification problem is constructed as an optimization problem, which can be solved by neural networks. In the following subsections, we will present Hopfield neural network based verification methods.

2.2.1 Geometrical Feature Extraction and Graph Formation

To obtain the required facial features, a binary image is obtained by using a proper threshold value. It is essential to find the proper threshold value to distinguish the eyes, nose and mouth from the other face regions. There are many available methods to calculate the optimal threshold value [15] [16]. The methods showed a good result detecting facial components such as eyes, noses, and mouth. In our early study [17], we developed an algorithm that detects corner points of the facial components. We use the corner points of each facial component for extracting facial features. Corner points are important since the information of the shape is concentrated at the points having high curvatures. From the corner points, we can extract useful features. They are a local feature (an angle between neighboring corners) and relational features (distances between the corners). These two features, which are invariant under transnational and rotational changes, are used for the robust description of shape of facial components. A graph can be constructed for an object model using corner points as nodes of the graph. Each node has a local feature as well as relational features with other nodes. For the matching process, a similar graph is constructed for the input image which may consist of one or several overlapped objects. Each model graph is then matched against the input image graph to find the best matching subgraph.

2.2.2 Hopfield Neural Networks for Face Verification

The Hopfield neural network is constructed by connecting a large number of simple processing elements (neurons) to each other. A two dimensional array is constructed to apply a face matching problem into a neural network as shown in Figure 2. The columns of the array label the nodes of a face model, and the rows indicate the nodes of an input face. Therefore, the state of each neuron represents the measure of match between two nodes from each graph. For example, a white neuron in Figure 2 represents a successful match between corresponding feature points between the model and input. A shaded neuron represents a mismatch between corresponding feature points between the model and input. In general, for the i^{th} node in the input image and the k^{th} node in the object model, the ik^{th} processing neuron located in the i^{th} row and ki^{th} column.

The output of the ik^{th} neuron is fed to the input of the jl^{th} neuron by connection of strength C_{ijkl}. In addition, each neuron has external inputs (an offset bias) of I_{ik} to its input. The states of the neurons can be expressed by U_{ik}, the outputs by V_{ik}, the

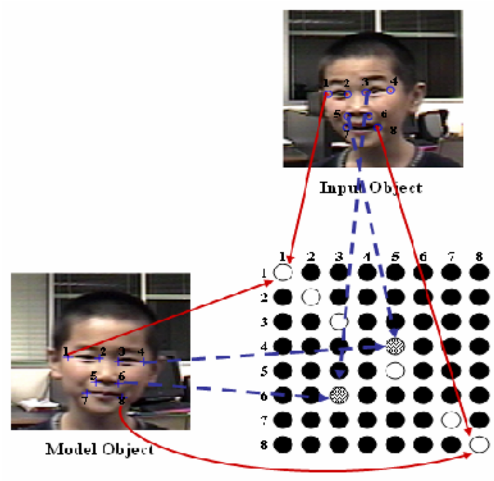


Fig. 2. 2-D array for Hopfield neural networks

connection strengths by C_{ikjl}, and the external inputs by I_{ik}. The matching process can be characterized as minimizing the following energy function:

$$
\begin{aligned}
E = -\frac{A}{2}\sum_i\sum_j\sum_k\sum_l C_{ijkl}V_{ik}V_{jl} \\
+\frac{B_1}{2}\sum_i\sum_k\sum_{l\neq k} V_{ik}V_{il} + \frac{B_2}{2}\sum_k\sum_i\sum_{j\neq i} V_{ik}V_{jl}
\end{aligned}
\tag{2}
$$

where V_{ik} is a binary variable which converges to "1" if the i^{th} node in the input image matches the k^{th} node in the object model; otherwise, it converges to "0". The first term in equation (2) is a compatibility constraint. Local and relational features which have different measures are normalized to give tolerance for ambiguity of the features. The last two terms are included to enforce the uniqueness constraint so that each node in the object model eventually matches only one node in the input image and the summation of the outputs of the neurons in each row or column is no more than 1. C_{ikjl} is normalized by a fuzzy function so that it helps us obtain good solutions. Therefore the coefficient A is supposed to be more emphasized in the matching problem. The comparability measure C_{ikjl} is expressed as follows:

$$
\begin{aligned}
C_{ikjl} &= W_1 \times F(f_i, f_k) + W_2 \times F(f_j, f_l) \\
&\quad + W_3 \times F(r_{ij}, r_{kl})
\end{aligned}
\tag{3}
$$

The fuzzy function *F(x, y)* has a value 1 for a positive support and -1 for a negative support. The value of *F(x, y)* is defined such that if the absolute value of the difference between x and y is less than a threshold, then *F(x, y)* is set to 1, otherwise *F(x,* y) is set to -1. The first term is related to the local feature of (*i, k*)th neuron. If the *i*th node of a model and the *k*th node of an input have similarity in their local features, then the value of the fuzzy function $F(f_i, f_k)$ is set to 1, otherwise set to 0. The second term is related to the local feature of (*j, l*)th neuron. $F(f_j, f_l)$ is set to 0 or 1 by the procedure as explained above. The third term is related to the relational feature between two neurons. If the relational feature between ith and *j*th node in a model is similar to the relational feature between *k*th and *l*th node, then $F(r_{ij}, r_{kl})$ is set to 1, otherwise set to 0. The coefficient W_i would add to 1. Therefore the value of C_{ikjl} is normalized from - 1 to 1. The performance of the algorithm is significantly influenced by the weight and the tolerance of the fuzzy function. As the tolerance 0 in the fuzzy function is larger, robustness of the algorithm is increased but mismatching may occur. On the other hand, as the tolerance θ is smaller, the algorithm becomes very sensitive to the noise level of an input so that a matchable node may not be detected. However, the mismatching rate will be decreased. For example, when the current comer detection algorithm is applied to a noisy image or a blurred image, comer points can be displaced by the smoothing effect. In this case, the tolerance should be increased even though it causes mismatching. The weight W_i is decided by the significance of the features. In this graph matching, local features do not contribute to interactions between neurons but relational features do. Therefore, relational features are more emphasized than local features in the neural network application, i.e. the weight of the relational feature W_3 has a larger value than other weights. The second term of the energy function of the matching problem is represented by a quadratic function so that it can be cast into the Hopfield energy function. The third term also has the same formation as the second term of the energy function. Therefore, equation (2) can be cast into a Hopfield style energy function as follows:

$$E = -\frac{1}{2}\sum_i \sum_j \sum_k \sum_l C_{ikjl} V_{ik} V_{jl} - \sum_i \sum_j I_{ik} V_{ik}$$

$$C_{ikjl} = AC_{ikjl} - B_1\delta_{ij} - B_2\delta_{kl} + (B_1 + B_2)\delta_{ij}\delta_{kl} \tag{4}$$

where $\delta_{ij} = 1$ when $i = j$, otherwise $\delta_{ij} = 0$. Hopfield proved that the energy function is a Liapunov function. Thus the energy function converges to a local minimum when the states of neurons converge to stable states. The matching process is based on global information of the image which provides excitatory or inhibitory supports for matching local features. The simulation is a random process which will arrive at a stable state when the energy function of equation (14) is at its minimum.

2.2.3 Three Face Reference Model

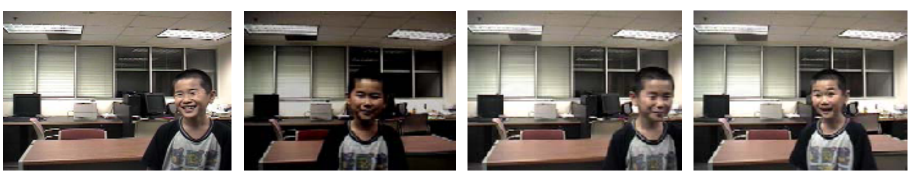
One of the challenging problems is recognizing a face with different poses. We use three face reference model (TFRM) that covers the frontal view, left view, and right view of a face. There are many advantages to using a TFRM for face verification. The TFRM covers all pose changes allowing for a better match for face verification. Different images from the database can be selected to represent the TFRM, allowing more comparisons, providing a more comprehensive matching process for face verification. The TFRM is robust; mismatching 2 out of 3 models do not affect overall performance. Face verification can still be obtained even if 2 of the 3 models do not match. If at least one reference model match, face verification is a success. The TFRM has a higher success rate than a reference model based on a single view because the image can be compared against three images with different angles of view instead of one single image. In our experiment, TFRM will be compared to SFRM.

3 Simulation Results

Our face verification method has been evaluated using a face image database containing 400 sequences of images. Each image contains the same face with variations in position, scale, and facial expression. Our algorithm verifies a face within an image by comparing it to a reference model. For face verification, images in the database will be used as input images and as reference models.

We use two methods to select images for face verification; face verification by comparing randomly selected images or face verification by selecting two neighboring images. For random comparison of images, we use the three face reference model (TFRM). It consists of three images of three different facial poses selected from our image database; frontal view, right view, and left view. If the input image matches at least one of these three images, verification is a success.

The following is a successful face verification example using a three face reference model (TFRM). In Figure 3, the facial image is compared to the frontal, right and left face views using the Hopfield neural network. Table 1 shows the matching results of the images. When the tolerance is set to 5, face verification fails when the 15th image in the database is compared to the frontal and right view of the TFRM; however, face verification is successful when compared to the left view. Face verification for this image is a success because it matches one of the TFRM.



(a) (b)

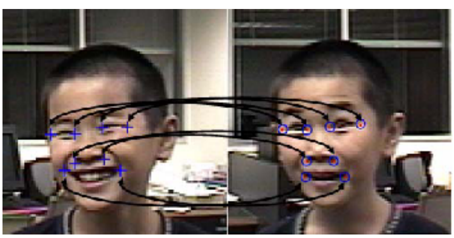
Fig. 3. (a) Input image (b) Three face reference model (TFRM): frontal view, right view, left view

Neural networks provide good matching results between a model and input image. Its algorithm is applied for matching feature points of the face using a 2-dimensional array (rows represent the model image, columns represent input image). Two points are used to detect each facial component; eyes, nose, and mouth, therefore, 8 feature points are used to recognize a face. Feature points 1 and 2 represent the corner points on each side of the left eye. Feature points 3 and 4 represent the corner points on each side of the right eye. Feature points 5 and 6 represent the corner points on each side of the nose.

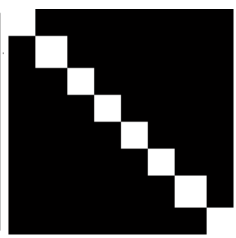
Feature points 7 and 8 represent the corner points on each side of the mouth. Each feature point in the model is compared to all feature points of the input image.

Figure 4 is an example of a successful and unsuccessful single model case for face verification. The following is a success case using a single model. Straight lines indicate the matching of feature points of similar facial components and dotted lines indicate the matching of feature points of dissimilar facial components. The eyes, nose, and mouth of a model image are compared to an input image by matching their feature points. All feature points match between the model and input image. Eight diagonal 1's are located in the (1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), and (8,8) position of the array indicating the matching of feature points between similar facial components. 0's in the array indicate other feature points in the image do not match. Face verification is a success because at least 5 feature points must match between corresponding components and this image has 8 matching feature points. The graphical image of the 2-dimensional array shows 8 diagonal white squares representing matched feature points and black squares representing unmatched feature points.

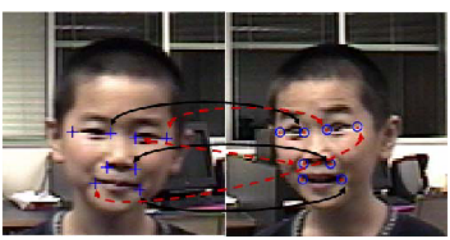
The following is an example of a single model case failure. The eyes, nose, and mouth of a single model image and an input image are compared by matching their feature points. Three pairs of feature points, 2 (right corner point of eye), 6 (right corner point of nose), and 8 (right corner point of mouth), match between the model and input image. A 2-dimensional array is used to represent matching of feature points. Three diagonal 1's are located in the (2,2), (6,6) and (8,8) position of the array indicating the matching of feature points between similar facial components. 1's located outside the diagonal, located in positions (3,5) and(7,4) indicates a matching of feature points of dissimilar facial components (i.e. (3,5) indicates matching of left corner point of eye and left corner point of nose). 0's indicate feature points in the image do not match. Face verification fails for this input image because at least 5 feature points must match



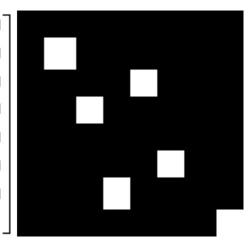
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



(a)



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



(b)

Fig. 4. (a) Success case (b) Failure case

between similar components, however, this image has only 3 matching corresponding feature points indicated by 3 diagonal 1's in the array. The graphical image of the 2-dimensional array shows 3 diagonal white squares indicating failure. Also, 2 white squares are located outside the diagonal indicating a matching of feature points of dissimilar facial components.

In Table 1, we present matching ratio results of the face verification method applied to the face image database using HNN. Matching ratio is defined as the ratio of the number of images that match a reference model to the total number of images input. Table 1 is used to measure the performance of two types of reference models using random images. A matching ratio is simulated on 17 images. With an optimum tolerance of 4, 16 out of 17 images match the single face reference model (SFRM) resulting in a matching ratio of 94.12%. However, 17 out of 17 images match the TFRM which has a greater matching rate of 100%. In Figure 5, the results show that the TFRM has a higher matching rate for face verification than SFRM.

Table 1. Matching Ratio using HNN (SFRM vs TFRM)

Tolerance	Matching Ratio	
	SFRM	TFRM
1	0.5294 (9/17)	0.5294 (9/17)
2	0.8824 (15/17)	0.9412 (16/17)
3	0.9412 (16/17)	0.9412 (16/17)
4	0.9412 (16/17)	1.0000 (17/17)
5	0.5294 (9/17)	0.8824 (15/17)
6	0.7059 (12/17)	0.7647 (13/17)

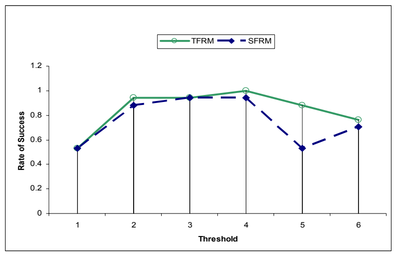


Fig. 5. Three face reference model (TFRM) using HNN

Fig. 6. Neighboring images in a sequence

For face verification by selecting neighboring images in a sequence (Figure 6), we select 2 frames in sequence. The first image is the reference model image and the neighboring image is the input image to be compared against the reference model. For each frame in the sequence, the current frame is used as a model and is compared to its neighboring input image. Figure 7 shows matching results for the neighboring sequence. The results show that with tolerances 3 and 4, a matching result was 100%. Using neighboring frame as a reference model yielded better results than using a fixed single face reference model. The advantage of face verification using neighboring images is that we can have a better success rate than a fixed reference model. However, it has a problem with a drift. Tracking will fail if a severe drift occurs due to the result of poor segmentation.

Extracting feature points is not reliable under unrestricted and poor lighting conditions. Some points may not be detected or may be off from true position. We created situations where a few feature points were missing or off from true position to test our neural network based verifiers. The first simulation uses an image missing two feature points from the same facial component, (i.e. a feature point missing on each side of the right eye, therefore, right eye not detected in the image). The second simulation uses an image missing two feature points, one from a different facial component, (i.e. one feature point missing from the right side of the left eye, one feature point missing from the left side of the nose). If four or more feature points of the image match the reference model, a successful match is obtained. The matching ratio results are based on 170 experiments.

After calculating the matching ratios for images missing combinations of feature points and images missing two feature points from the same facial component, the average matching ratio results were calculated. Table 2 presents the HNN matching

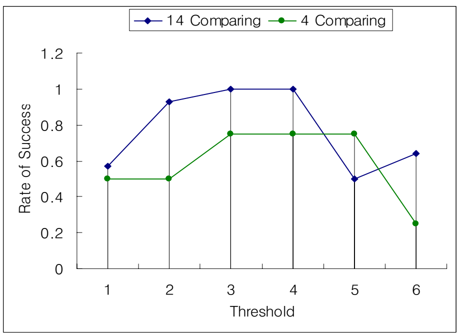


Fig. 7. Neighboring sequence

Table 2. Measures HNN average performance

Tolerance	Matching Ratio	
	SFRM - HNN	TFRM - HNN
1	0.3628	0.3799
2	0.6155	0.6544
3	0.6253	0.7059
4	0.6863	0.7451
5	0.5907	0.7427
6	0.5651	0.5651

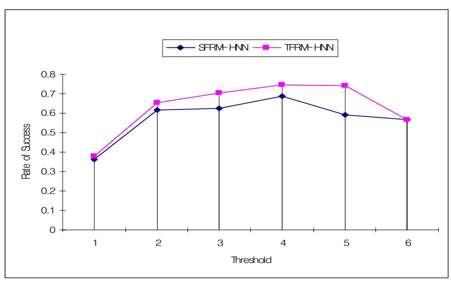


Fig. 8. Compares HNN average performance of SFRM and TFRM

ratio averages for the SFRM and the TFRM. With an optimum tolerance set at 4, the SFRM results in an average matching ratio of 68.63%, and the TFRM results in a higher average matching ratio of 74.51%. Comparing the SFRM to the TFRM and applying the HNN, the TFRM averages are constantly higher (Figure 8).

In the third simulation, two feature points are forced to locate off the true positions. This image is referred to as a poorly segmented image. For face verification and recognition, a comparison is made between each of the 8 feature points in the reference model and the 8 feature points in the poorly segmented image. The results of the simulation show images with changed features points have a lower matching ratio than original images; images with feature points located at original positions.

Tables 3 measures the HNN matching ratios for original images and poorly segmented images, respectively. When using a TFRM, both images have a 100% matching ratio when the tolerance is set at 3 and 4. In general, higher matching ratios result when both types of images are compared to the TFRM. Both images are compared to the SFRM in Figure 1 and the TFRM in Figure 2. The graphs indicate matching original images to a TFRM results in higher matching ratios.

Table 3. Measures HNN performance using points off from true positions

Tolerance	HNN-Original		HNN-Changed Points	
	SFRM - HNN	TFRM - HNN	SFRM -HNN	TFRM - HNN
1	0.8	0.8	0.6	0.8
2	0.8	0.8	0.6	0.6
3	1	1	1	1
4	0.6	1	0.6	1
5	0.6	1	0	0.4
6	0.2	0.4	0	0.2

Yet again, comparing original images to TFRM achieve the best results. Overall, HNN show good performance under poor pre-processing conditions such as mission points or false detection of points. Algorithms have been implemented in Matlab. Experiments are done to evaluate the efficiency of our face verification method under various conditions, such as, changes in the types of reference models. The performance measure used is a matching ratio which verifies the excellent performance of our face verification method. [9]

4 Conclusion

This paper presented a robust face verification method that utilized geometrical facial information and neural networks. If multiple visual modalities such as gradient intensity, color, and geometrical features, are combined well, we could expect more robust results. A HNN based verifier has been presented. It shows good performance in terms of matching ratios. Our simulation results show that our face detection approach performed well using a three face reference model. Our technique can be extended to face recognition without modification of neural network based verification.

References

[1] Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, January 2002

[2] J. Yang, A. Waibel, A real-time face tracker, In Proc. of WACV'96, (1996) 142-147.

[3] S. Birchfield, Elliptical Head Tracking Using Intensity Gradients and Color Histograms, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA pp 232-237, June 1998.

[4] Szu-Hao Huang, and Shang-Hong Lai, "Detecting Faces from Color Video by Using Paired Wavelet Features", *Computer Vision and Pattern Recognition Workshop*, 27-02, pp. 64-64, Jun. 2004

[5] Y.Rodriguez, F.Cardinaux, S.Bengio, and J. Mariethoz, "Estimating the quality of face localization for face verification", *Proceedings of International Conference on Image Processing (ICIP '04)*, vol.1, no.22-27, pp. 581-584, Oct. 2004

[6] Ming-Jung Seow, D.Valaparla, and V.K. Asari, "Neural network based skin color model for face detection", *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, pp.141-145, Oct. 2003

[7] Q.Yang, X.Ding, and Z.Chen, "Discriminant local feature analysis of facial images", *Proceedings of International Conference on Image Processing (ICIP '03)*, vol 2, pp. II - 863-6 vol.3, Sept. 2003

[8] T. Darrell, et. al., "Integrated person tracking using stereo, color, and pattern detection," *Conference on Computer Vision and Pattern Recognition, Santa Barbara*, CA, 601—608, June 1998.

[9] Gwang M. Kim, DongCheng Lin, Jung H. Kim, and Sung H. Yoon, "Robust Head Tracking Based on Hybrid Color Histogram and Random Walk Kalman Filter," *In The Multi-modal Speech Recognition Workshop*, Greensboro, NC, June 2002.

[10] R.P. Lippman, "An introduction to computing with neural networks," *IEEE ASSP*, pp4-22, 1987.

[11] M. Takeda and J. W. Goodman, "Neural networks for computation: Number representations and programming complexity," *Applied Optics,* (25) 18, Sept. 15, 1986.

[12] S. German, D. German, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6,* 721-741, 1984.

[13] G. Bilbro, W. Snyder, "Applying mean field annealing to image noise removal," *Journal of Neural Network Computation*. pp5-17, 1990.

[14] E. Van Den Bout, T. K. Miller III, "Graph partitioning using annealed neural networks, " *IEEE Trans. Neural Networks* 1 (2192-203, 1990.

[15] Li Zhuang, Guang-you Xu, Haizhou Ai, and Gang Song, " Detection and tracking of facial features under a complex background" *Human Vision and Electronic Imaging* VII, June 2002, pp. 448-454.

[16] T.C. Chang, T.S. Huang, and C. Novak, "Facial feature extraction from color images", *Proc. the 12th IAPR International Conference on Pattern Recognition*, Vol. 2, pages 39-43, 1994

[17] Gwang Kim, Jung H. Kim, Sung Ho Yoon and Gi Taek Hur, "Lip Boundary Detection Techniques Using Color and Depth Information", Proceedings of SPIE 2002 International Symposium on Electronic Imaging, San Hose, CA, USA.

[18] Chow, T.Y. and Lam, K.M., "Mean-shift based mixture model for face detection in color image" *Proceedings of International Conference on Image Processing (ICIP '04)*, pp601-604, 2004

[19] V.V. Starovoitov1, D.I Samal1, D.V. Briliuk1, Three Approaches for Face Recognition", *The 6-th International Conference on Pattern Recognition and Image Analysis*, Russia, pp. 707-711 October, 2002