

An Annealed Neural Network for Reliable Face Detection

Sung H. Yoon^a, Giyeon Park^b, Ji Hyun Lee^b, Jung H. Kim^b

^aComputer Science Dept., North Carolina A&T State University, NC 27411, USA

^bElectrical Engineering Dept., North Carolina A&T State University, NC 27411, USA

Abstract

This paper presents an annealed neural network based method for face detection. We present a robust algorithm that improves face detection and tracking in video sequences by using geometrical facial information and an annealed neural network 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: annealed neural network, face detection

1. Introduction

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 [1-6]. 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.

We use geometric feature points for face verification based on neural networks. The

geometric feature 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.

Under heavily occluded conditions, Hopfield neural network (HNN) may have feature points mismatched or unmatched, and as a result it doesn't guarantee a global optimal solution because it is basically depending on its initial states. A HNN can solve optimization problems faster than any other algorithms when an energy function is given correctly. It does generally guarantee a local minimum, but not global.

The Annealed Hopfield Network (AHN) has been developed to provide near global solutions without initial restrictions and less false matching than HNN. The AHN is presented to find a robust solution for face verification and recognition problems in a video sequences. To improve the capability of detecting corners, another approach to boundary smoothing for curvature estimation is proposed. This method is based on a minimization strategy known as mean

field annealing (MFA). To improve the computational complexity of the AHN and provide reliable matching under occluded conditions, the mean field theory is applied to the Hopfield network. We can detect corners easier and better in this approach than the constrained regularization approach which may result in loss of corners.

The relationship between the HNN and MFA has been shown by Van Den Bout [7]. The evolution of a solution in a HNN is a special case of the relaxation toward a stable state affected by MFA at a fixed temperature T [8, 9]. He has shown that there is a correspondence between the temperature T in MFA and the neural gain λ in the HNN. In this paper, the AHN is described for face verification and recognition in a multi-context scene using the advantages of MFA.

2. Theoretical concept of AHN

A motion equation is shown in Eq. (1) with the sigmoid function g in Eq. (2).

$$\frac{du_{ik}}{dt} = -\frac{u_{ik}}{\lambda} + \sum_j \sum_l T_{ijkl} + I_{ik} \quad (1)$$

$$g(u_{ik}) = \frac{1}{1 + e^{-u_{ik}/\lambda}} \quad (2)$$

Our energy function of the matching problem is organized as Eq. (3).

$$E = -\frac{1}{2} \sum_i \sum_j \sum_k \sum_l T_{ijkl} V_{ik} V_{jl} - \sum_i \sum_k I_{ik} V_{ik} \quad (3)$$

The output of each neuron for the matching problem has the value of 0 or 1 to represent measure of similarity. We will call the output of each neuron a spin for the MFA approach. That is, each neuron in Figure 1 is substituted by a spin. It was assumed that the *spin* interactions C_{ijkl} are symmetric and have no self-interaction

(i.e., $C_{ijkl} = 0$). The state space of each *spin* is [7]:

$$s_{ik} \in \{0,1\} \text{ for } 1 \leq i, k \leq N \quad (4)$$

where $N^2 = m \times n$ in the m by n two dimensional array. In the simulated annealing, the random perturbations move the system towards its thermal equilibrium at the current temperature. Assuming that all the *spins* are at equilibrium, one can determine the equilibrium *spin* average of the ik th *spin* $\langle s_{ik} \rangle$ from the Boltzmann distribution and the change in the average system energy as s_{ik} flips from 0 to 1.

For an illustration, let

$$H_0 = \langle H(s) \rangle_{s_{ik}=0} = 0 \text{ and } H_1 = \langle H(s) \rangle_{s_{ik}=1} = 1 .$$

Since the system is Boltzmann distributed, the equilibrium value of $\langle s_{ik} \rangle$ is calculated as follows:

$$\begin{aligned} \langle s_{ik} \rangle &= \Pr\{s_{ik} = 0\} \times 0 + \Pr\{s_{ik} = 1\} \times 1 \\ &= \frac{\exp(-H_1/T)}{\exp(-H_0/T) + \exp(-H_1/T)} \\ &= \left\{ 1 + \exp\left[-\frac{(H_0 - H_1)}{T}\right] \right\}^{-1} \\ &= \left\{ 1 + \exp\left[\frac{u_{ik}}{T}\right] \right\}^{-1} \end{aligned} \quad (5)$$

We define u_{ik} to represent the quantity $H_0 - H_1$, which is the mean or effective field experienced by the ik th *spin*. Unfortunately, it is in general difficult to compute u_{ik} for large N :

$$\begin{aligned} \langle H(s) \rangle &= \sum_i \sum_j \sum_k \sum_l \langle C_{ijkl} s_{ik} s_{jl} \rangle + \sum_i \sum_k \langle I_{ik} s_{ik} \rangle \\ &= \sum_i \sum_j \sum_k \sum_l C_{ijkl} \langle s_{ik} s_{jl} \rangle + \sum_i \sum_k I_{ik} \langle s_{ik} \rangle \end{aligned} \quad (6)$$

The difficulty arises from the fact that s_{ik} and s_{jl} are not independent, so that their expected values are not separable in the above equation. However, when the number of interacting spins is large enough that the effect of any single *spin* on any other *spin* is very small in comparison to the total field, then the mean field approximation can be used:

$$\langle H(s) \rangle = \sum_i \sum_j \sum_k \sum_l C_{ijkl} \langle s_{ik} \rangle \langle s_{jl} \rangle + \sum_i \sum_k I_{ik} \langle s_{ik} \rangle \quad (7)$$

Eq. (5) and Eq. (7) has the same structure as Eq. (2) and Eq. (3). The spin interaction C_{ijkl} as a comparability measure in the Hopfield neural networks is expressed as follows:

$$C_{ijkl} = W_1 \times F(f_i, f_k) + W_2 \times F(f_j, f_l) + W_3 \times F(r_{ij}, r_{kl}) \quad (8)$$

Local and relational features which have different measures are normalized to give tolerance for ambiguity of the features. The fuzzy function F 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 θ , then $F(x, y)$ is set to 1, otherwise $F(x, y)$ is set to -1. The sum of the weights is 1. We use two features, *angle* as a local feature at the corner point in the object boundary and *distance* as a relational feature between the corners. Angle helps us to recognize the shape of object. However, false segmentation causes to generate different angles from those of original segmentation. In this paper, relational features are more emphasized than local features. AHN even works well without local features.

In addition, random perturbation to move the system towards its thermal equilibrium in simulated annealing is the same as updating rule of the Hopfield network. The

only difference is that λ in Eq. (2) is replaced with temperature T in Eq. (5). It means that given T , the flow to thermal equilibrium in MFA is the same as the flow of Hopfield network given T . Therefore, if we find the stable points of states by slowly lowering λ from the high value, then we will find global solutions or near global solution of the network without initial restriction.

3. Feature extraction and graph formation

Our algorithm transforms the original image (RGB format) into $YCbCr$ images; using the Y channel to get an edge image; using C_b and C_r channels to get the skin tone image. The skin tone image and the edge image are processed into a skin tone edge image. We combine the skin tone image and edge image to get a good face candidate and to remove background noise. Background noise must be eliminated because any pixel can have a skin color range which can result in the background having skin color like a face. After applying the morphological operation using erosion and dilation on the skin tone edge image, a large number of small skin color regions are eliminated. Erosion and dilation methods are used to identify homogenous areas and to remove noises to get a binary edge image. In order to detect potential face candidates, we set the threshold value to convert the skin tone edge image into a binary image. Based on the binary image, a graph is used to detect face candidate. We use the edge image to determine the eyes and mouth position using an ellipses model to get the feature point positions.

4. Hopfield neural networks for face verification

The Hopfield 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 matching problem into a neural network as shown in Figure 1. The columns of the array label the nodes of an object model, and the rows indicate the nodes of an input object. Therefore, the state of each neuron represents the measure of match between two nodes from each graph. 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 k th column is described by two variables which are current state and output. The current state and output are generally denoted by u_{ik} and v_{ik} , individually. The output is usually related to the state by simple nondecreasing monotonic output function $g(u_{ik})$. This function (normally nonlinear) is designed to limit the possible values of v_{ik} to the range -1 to +1. In other words, $g(u_{ik})$ is frequently a step function (in the case of discrete Hopfield network) or a sigmoid function (in continuous Hopfield network). The output of the ik th neuron is fed to the input of the jl th neuron by connection of strength T_{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 connection strengths by T_{ijkl} , and the external inputs by I_{ik} .

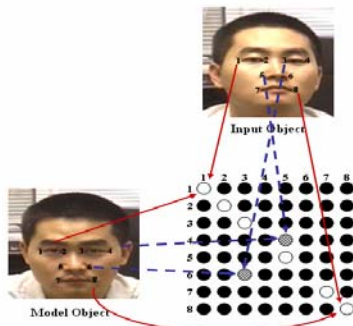


Figure 1: 2-D array for Hopfield neural networks. White neuron = match between corresponding features of model and face image. Black neuron = no match between corresponding features of model and face image. Shaded neuron = match, but match between non-corresponding features.

5. Simulation results

Our face verification simulation involves a face image database containing video sequences of images. Each image contains the same face with variations in position, scale, and facial expression. Variations in facial views increase the accuracy of our face verification algorithm. Our algorithm uses reference models to verify a face within an image. Images in the database will be used as input images and as reference models. Input images are compared to reference models using AHN to confirm that the face in the input image matches the face in the reference model.

A set of 23 sequences of images are picked, from our database of 1377 video sequences of images. 3 images, each with different facial views are picked as reference images and 20 images are used as input images. We set up different tolerance ranges for our experiment. The tolerance ranges from 1 to 6. Each input image is compared to the TFRM (three images), 6 times, once for each tolerance. 360 experiments (comparisons) are conducted using the TFRM. In addition, each input image is compared to the SFRM (one image), 6 times, once for each tolerance. We performed 120 experiments using the SFRM. A total of 480 experiments are conducted. A matching ratio is simulated on the 20 input images. Matching ratio is defined as the percentage calculated by dividing the total number of successful matches between an input image and reference model by the total number of images input.

AHN provides good matching results between 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. Each feature point in the model is compared to all

feature points of the input image. If 5 - 8 feature points match between model and input image, a perfect match is obtained. If 1 - 4 feature points match, face verification between the model and input image fails.

Face detection verification is also measured using the neural network AHN. We used an AHN algorithm to measure its matching performance. AHN is used to find a robust solution for occluded object matching. The same procedure was performed as HNN, however, we replaced HNN with AHN. The robustness of our AHN algorithm is proved by identifying faces with variations in the facial poses.

The AHN simulation results of the algorithm involving original images are shown in Table 1. In most cases, the TFRM has a higher success rate than the SFRM. Figure 2 shows matching ratios of SFRM and TFRM. The results show matching using the TFRM is successfully obtained.

As shown in Table 2 and displayed in Figure 3, the AHN matching ratios related to images with missing points yields a high matching ratio when applying the TFRM. The face detection and tracking improves better than HNN, when our algorithm uses AHN.

The AHN matching ratios for poorly segmented images, images with points off from true positions, are given in Table 3. The TFRM achieves high ratios. These results are displayed in Figure 4 which verifies the excellent performance of the algorithm. When comparing the performance of the HNN with that of AHN in finding faces with their feature points off from true positions, AHN shows better performance.

6. Conclusion

This paper presents experiments done to evaluate the efficiency and verify the excellent performance of our face

verification and recognition method under various conditions. A robust verification method has been presented which utilizes geometrical facial information and neural networks. Our simulation results show that our face detection approach performed well using a three face reference model. We present a HNN based verifier, however, face detection and tracking improves when our algorithm uses the neural network based verifier, AHN. Our results show that AHN is more reliable, powerful and have a higher success rate for image matching than HNN. The matching ratios indicate AHN's matching performance is superior to that of HNN.

Tolerance	Matching Ratio	
	SFRM	TFRM
1	0.0500(1/20)	0.1500(15/20)
2	0.4500(9/20)	0.8000(18/20)
3	0.7500(15/20)	1.0000(20/20)
4	0.9500(19/20)	1.0000(20/20)
5	1.0000(20/20)	1.0000(20/20)
6	0.9000(18/20)	1.0000(20/20)

Table 1: Matching Ratio using AHN (SFRM vs TFRM)

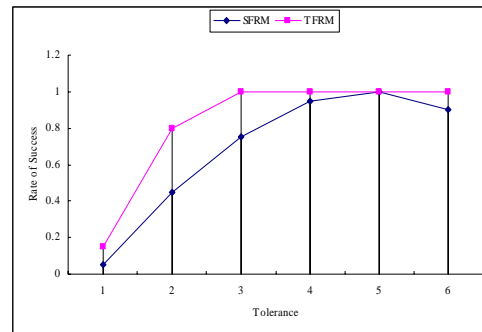


Figure 2: AHN performance of SFRM and TFRM

Tolerance	Matching Ratio	
	SFRM	TFRM
1	0.0000(0/20)	0.0000(0/20)
2	0.3000(6/20)	0.6000(12/20)
3	0.4500(9/20)	0.8000(16/20)
4	0.8500(17/20)	1.0000(20/20)
5	0.8500(17/20)	0.9000(18/20)
6	0.8000(16/20)	0.9000(18/20)

Table 2: Measures AHN Missing points

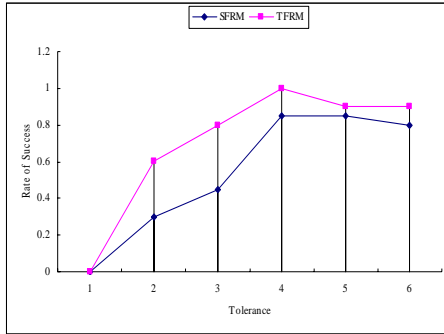


Figure 3: Compares AHN performance of Missing Points

Tolerance	Matching Ratio	
	SFRM	TFRM
1	0.1000(2/20)	0.3000(6/20)
2	0.5500(11/20)	0.7000(14/20)
3	0.7500(15/20)	0.9500(19/20)
4	0.9500(19/20)	1.0000(20/20)
5	0.9500(19/20)	0.1000(20/20)
6	0.8500(17/20)	0.9500(19/20)

Table 3: Measures AHN performance using points off from true positions

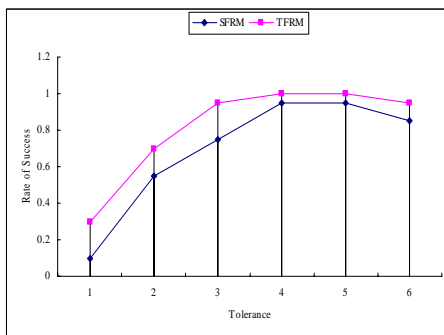


Figure 4: Compares AHN performance using points off from true positions

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*, pp. 142-147, 1996.

[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, June 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] E. Van Den Bout, T. K. Miller III, Graph partitioning using annealed neural networks, *IEEE Trans. Neural Networks 1 (2)*, pp. 192-203, 1990.

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

[9] G. Bilbro, W. Snyder, Applying mean field annealing to image noise removal, *J. Neural Network Computing*, pp. 5-17, 1990.