Eye Shape Detection in Log-Polar Domain Using Recurrent Neural Network

Mohammad Aldibaja*, Shinichi Suzuki

Department of Mechanical Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan

*maddib@yahoo.com

Abstract

This paper presents a method to detect the eye shape using recurrent neural network. The eye image is decomposed into a series of eye profiles using Log-polar transform. An eye profile represents the change of pixel values along Log r coordinates at an angle θ . Each eye profile contains a pixel that represents the eye contour. By considering the eye profiles as a time series, a recurrent neural network is trained to predict the locations of the eye contour. The experimental results verified that the recurrent network is able to detect the eye shape under different lighting conditions, scales and rotation angles. Moreover, the proposed method outperforms the conventional methods in accuracy and detection time.

Keywords: Eye Feature Extraction, Object Shape Detection, Recurrent Neural Network, Log-Polar Transform.

1. Introduction

Human-machine interface systems have become one of the most attractive fields in image processing^(1,2). The common approach is to use image processing techniques to detect human interactive-components (hand, face, mouth, eyes...etc) and then extract the components' features. Eye is considered as a very promising component to be used in interactive systems recently. It has been proved that using eyes in interface systems is faster than hands⁽³⁾. Eye features such as eye corners, gaze point and blinking frequency can be employed in many applications. Nowadays, Car companies try to develop driving system to be interconnected with eyes. For example, the driver fatigue can be detected by tracking the blinking frequency⁽⁴⁾. Car accidents can be avoided by observing whether the driver has seen road events such as traffic signals, road signs and pedestrian crossing the street⁽⁵⁾. On the other hand, the

handicapped people who have problems in controlling wheel chair by hand can be helped using eye gaze detection techniques⁽⁶⁾.

Many methods have been proposed for object shape detection in the literature⁽⁷⁻¹⁰⁾. For example, Active shape model (ASM) is one of the most famous and simplest techniques that can be used for eye shape detection⁽⁷⁾. Active Appearance Model (AAM) has been suggested as an advanced version of ASM using object shape and texture models⁽⁸⁾. Yuile *et al.* has proposed Deformable Template (DT) to detect the eye shape⁽⁹⁾.

Most of these methods suffer from the difficulties of detecting the eye shape under different rotation angles, scales, lighting conditions and translations. Moreover, the essence of these algorithms is to detect the eye shape using an iteration technique. Therefore, the number of iterations is not fixed and differs from an eye image to another. Thus, these methods are not appropriate for real time applications such as interactive systems.

In this paper, the proposed method is inspired by our previous work⁽¹¹⁾. The eye shape is represented by a set of points. The distribution of eye shape points represents lower and upper eyelids and two eye corners as well. The eye image is transformed into Log-polar domain (LPT) in order to make the proposed method scale-rotation invariant⁽¹²⁾. Each row in the transformed image is called an eye profile at corresponding θ . Every eye profile contains a part of the eye contour. A recurrent neural network is trained to model and analyze the patterns of pixel value distributions along the eye profiles. Thus, the learned network tries to predict the location of the eye contour in an eye profile using some inputs and some previous output conditions. The experimental results showed a better performance of the proposed method compared to ASM and DT.

The reminder of this paper is organized as follows: In

section 2, we provide a brief explanation on eye representation in Log-polar domain. Section 3 describes the structure of the recurrent neural network in order to detect the eye shape. Setup and experimental results of the proposed method are declared in section 4. The results are discussed in section 5 and we then conclude the essence of the proposed method with highlighting the yielded advantages in section 6.

2. Eye in Log-Polar Domain

2.1 Log-Polar Transform

An image in Cartesian coordinate system is exponentially sampled around the image center. Meanwhile, the eye image is radially divided into sectors⁽¹²⁾. Therefore based on the sector's radius and angle, a pixel $I = [x, y]^T$ is transformed into Log-polar domain $\hat{I} = [r, \theta]^T$ using Eq. (1). The width of the transformed image determines the number of ring sampling rate whereas the height tells the number of sectors.

$$\hat{I}_{\log r, \theta} = \begin{bmatrix} \log r = \log \sqrt{(x^2 + y^2)} \\ \arctan \frac{x}{y} \end{bmatrix} \quad (1)$$

2.2 Eye representation in Log-Polar Domain

Figure 1 shows a set of eye images in Cartesian coordinate system and corresponding Log-polar transformed images. The eye images are captured under different scales and rotations. The differences are transformed into translations along θ and $\log r$ axes in Log-Polar domain. One can observe that the eye contour exists at every θ . Therefore, detecting eye shape points is simplified to search on $\log r$ coordinates of the eye contour instead of searching on (x,y) coordinates in Cartesian coordinate system.

In the Log-polar transformed image, each row (eye profile) represents a pattern of pixel values distribution. The eye profile can uniformly be classified into two sub-segments. The first sub-segment may contain sclera, iris and pupil hence the corresponding pixel values are approximately low. The second sub-segment represents flesh and may contain eyebrows and the corresponding pixel values are high. The areas of these two sub-segments change according to the eye topological structure and θ . The eye contour can be considered as the intersection point between the two sub-segments in each eye profile. Therefore, an eye shape point can be extracted by searching

Fig. 1. Eye image in Cartesian and Log-polar coordinate systems.

on this point. In order to achieve this task, a recurrent neural network is used as detailed in the next section.

3. Eye Shape Points Extraction

3.1 Recurrent Neural Network

Neural network can be classified into recurrent and non-recurrent networks⁽¹³⁾. In non-recurrent network, the output is calculated by propagating the input values through the feedforward connections. On the other hand, the recurrent networks use the previous state of outputs as a part of the current input. This makes recurrent networks more powerful to model time series signals⁽¹⁴⁾ and more robust in terms of generalization and prediction⁽¹⁵⁾.

The recurrent neural network consists of a set of neurons that are fully or partial connected to each other. The connections are weighted by weights w that express the strength between the neurons. Each neuron receives inputs from the other neurons and transmits its output to the others as well.

In order to train a recurrent network to model a function, the weights must be adjusted using a training database and learning technique. The well-known learning technique called back propagation is used to predicate a one change ahead in the eye profile⁽¹⁶⁾.

Figure 2 shows the general structure of the used recurrent network. It consists of three layers; input, hidden and output. The input layer consists of one current input and I external previous inputs. The neurons in the hidden and output layers are fully connected. The output layer has one neuron without any direct connection to the input layer.

In order to train the network, the error E between the desired output D and the actual network output O must be minimized, Eq. (2). This can be optimized by adjusting the

Fig. 2. Recurrent neural network structure.

network weights using gradient descent method and Chain rule⁽¹⁶⁾, Eq. (3).

$$\frac{1}{2}E^2(t) = \frac{1}{2}[D(t) - O(t)]^2 \quad (2)$$

$$\Delta w = -\eta \frac{dE(t)}{dw} \quad (3)$$

where η is the learning rate which determines the descent step on the error surface. The term $dE(t)/dw$ represents the output sensitivity to a small change in w at time t .

In order to update the neurons' weights, we unify the representation of neuron terminals by y_k . Thus, I denotes current/external inputs in , C denotes bias b and U denotes the neuron output O , as follows:

$$y_k(t) = \begin{cases} in_k(t) & \text{if } k \in I \\ b = -1 & \text{if } k \in C \\ O_k(t) & \text{if } k \in U \end{cases} \quad (4)$$

Using the current conditions of weights, the output of a neuron is calculated as follows:

$$O_k(t) = f(net_k(t)) \quad (5)$$

$$net_k(t) = \sum_{l \in U \cup I} w_{kl} y_l(t) \quad (6)$$

where $f(\cdot)$ is the output activation function. The activation function has a significant role to model the nonlinearities of pixel values in the eye profiles. Therefore, the standard sigmoid function is used to represent the neurons' outputs. Thus, the gradient step of a neuron output is given by Eq. (7).

$$\frac{\partial O_k(t)}{\partial net_k(t)} = O_k(t) * (1 - O_k(t)) \quad (7)$$

Notice that the neural network output is one step

ahead $O(t+1)$. This means that the network is trained to predict the next future output based on the inputs. The sensitivity of a neuron output to a small change in weights of its input connections is calculated as follows:

$$\begin{aligned} \frac{\partial O_k(t)}{\partial w_{ij}} &= \frac{\partial O_k(t)}{\partial net_k(t)} \frac{\partial net_k(t)}{\partial w_{ij}} \\ \frac{\partial net_k(t)}{\partial w_{ij}} &= \sum_{l \in U \cup I} w_{kl} \frac{\partial y_l(t-1)}{\partial w_{ij}} + \delta_{ik} y_j(t) \end{aligned} \quad (8)$$

where δ_{ik} is the Kronecker delta function and equals to one if and only if $i=k$, otherwise it is zero. The term $w_{kl} * \partial y_l(t-1)/\partial w_{ij}$ represents explicitly the effect of changing w_{ij} on the other neurons.

On the other hand, the inputs and bias don't depend on the network weights. Therefore, this term takes the following values:

$$\frac{\partial y_l(t-1)}{\partial w_{ij}} = \begin{cases} 0 & \text{if } l \in I \text{ or } C \\ \frac{\partial O_l(t-1)}{\partial w_{ij}} & \text{if } l \in U \end{cases} \quad (9)$$

Finally, the weights of the hidden neurons are updated using Eq. (10) whereas those of the output neuron are updated by Eq. (11).

$$\begin{aligned} \Delta w_{ij} &= -\eta \frac{dE(t)}{dO(t)} \frac{dO(t)}{dnet(t)} \frac{dnet(t)}{dO_k(t)} \frac{\partial O_k(t)}{\partial w_{ij}} \\ &= \eta E(t) O(t) (1 - O(t)) w_{ij} \frac{\partial O_k(t)}{\partial w_{ij}} \end{aligned} \quad (10)$$

$$\Delta w_{ij} = -\eta \frac{dE(t)}{dO(t)} \frac{dO(t)}{\partial w_{ij}} \quad (11)$$

3.2 Eye Profile Modeling

Figure 3 illustrates some eye profiles taken at different θ . It emphasizes the uniform character of the eye profile with highlighting some differences according to the change of the topological eye structure. It can be observed that the pixel values of eye profiles vary smoothly and continuously. Moreover, the difference between two eye profiles at θ and $\theta+1$ is very small. Therefore, an eye profile can be approximately considered as a signal of time series. This signal can be analyzed and modeled in order to forecast the one future value. On the other hand, the eye profile model should have the capability to deal with noise. Noise may exist because of pupil location, red nerves in sclera, light reflectance and so on.

Each eye profile contains a pixel of the eye contour. The location of this pixel is the intersection point between the two sub-segments, flesh and eyeball, as explained early.

Fig. 3. Different eye profiles.

In order to detect the location of this point, we use the recurrent neural network to model the pattern of pixel values in the eye profiles.

In the learning stage, the pixel values of the eye contour in an eye image are changed to take a fixed value. As long as the pixel values in an eye profile change smoothly, the modified pixel value of the eye contour will appear in a pulse with a rapid change. The recurrent network is trained to predict one pixel value ahead based on the current and I previous inputs as well as the previous output of each neuron. Therefore if the eye profile without any modification is presented, the recurrent network is expected to predict ahead the location of the eye contour by outputting a sharp change of pixel values before one step of reaching the actual eye contour location. Hence, an eye shape point can be extracted accordingly. Repeating this operation in an eye image for many eye profiles according to a fixed sampling rate of θ , a set of eye shape points are obtained. These points represent the eye shape.

4. Experimental Results

4.1 Setup

The proposed method, ASM and DT have been implemented in the environment of visual studio C++ 2008. Matlab has also been used to create some graphs.

Two databases have been used to evaluate the performance of the methods. The first database includes 250 eye images from the UBIRIS database⁽¹⁷⁾, Fig. 4(a). The second database is locally collected to permit a considerable change of eye rotation with different scales and lighting conditions. The local database contains 100

Fig. 4. Databases.

right eye images as shown in Fig. 4(b). The eye shape of each image in the databases is described by $H=16$ points. These points are manually labeled over each image in Cartesian and Log-polar coordinate systems.

The recurrent network has the order of neurons of 5-20-1 for input, hidden and output layers; respectively. The activation function of a neuron output is standard sigmoid function. The θ sampling rate is $S=22$. Therefore, each eye profile is 1×256 pixels and each image is represented by a series of $H=16$ eye profiles. Consequently, this series consists of 16×256 pixel values.

In training stage, we used 50 eye images to train the recurrent neural network during 5000 epochs. Each eye image is converted into grayscale, rescaled between 0 and 0.9 and then transformed into a series of eye profiles according to the θ sampling rate S . The pixel values at the locations of the eye contour, obtained by manually labeling operation, have been modified to take 0.8. It has not been changed to 1 to avoid the saturation of the activation function that may lead the network to be unstable.

In detection stage, once a new image is given, the image is converted into grayscale, rescaled to the range [0-0.9] and Log-polar transform is applied. The Log-polar transformed image is then converted into a series of H eye profiles according to the fixed sampling rate S . The recurrent network then predicts the eye contour locations in these profiles.

In ASM, 16 points are used to represent the eye shape in Cartesian coordinate system. 150 manually labeled eye shapes have been used to model the dynamical change of the eye shape using Principal Component Analysis (PCA)⁽⁷⁾.

In DT, the eye shape is described by two parabolas; the upper and lower eyelids. The two parabolas are sampled into 16 points to measure the accuracy. Four potential field images have been used to highlight the eye features, valley image highlights the iris, edge image highlights boundary of iris and eye contour, peak image highlights sclera areas

(a) Prediction of the same training eye profile $\theta=359$.

(b) Prediction of different eye profile $\theta=355$.

Fig. 5. Eye contour detection using the trained recurrent neural network.

and gray image which highlights the brightness inside iris (light reflectance); respectively.

4.2 Results

Figure 5(a) shows an eye profile, green signal, taken from the eye image in Figure 3 at $\theta=359$. The signal is modified at the eye contour location to have a pulse with value of 0.8. The modified signal is used to train the recurrent network in order to predict one future step. Once learning operation is done, the raw signal is presented to the trained network. The network predicts one value ahead and generates a pulse at the location of the eye contour.

Figure 5(b) shows the eye profile at $\theta=355$ of the same eye image. This profile is presented to the trained network and a predicted signal is generated. The predicted signal possesses a sharp pulse that expected to indicate the location of the eye contour in the eye profile.

Figure 6(a) shows a series of $H=16$ eye profiles. These eye profiles represent an eye image sampled according to the sampling rate $S=22$. The modified series is modified to have sharp pulses at the actual locations of the eye contour in the image. The series is considered as a training signal

(a) Prediction of the same training eye image.

(b) Prediction of different eye image.

Fig. 6. Eye images represented by 16 eye profiles and eye contour detection using the trained recurrent network.

and used to train the network. After training, the raw series, which has no sharp pulses, is presented and a predicted signal is obtained. A new eye image has been presented to the trained network to detect the eye contour locations as shown in Fig. 6(b).

Figure 7 shows some eye images taken from the database. The corresponding exact eye shapes are manually labeled by $H=16$ points as illustrated in red points. The predicted eye shapes by the trained recurrent neural network are shown in green points.

Table 1 concludes the results of eye shape detection using the proposed method, ASM and DT. 300 eye images have been randomly taken from the two databases for this experiment. The table includes three comparative components; detection time, accuracy obtained and number of iterations.

5. Discussion

The eye profile is simple and its nonlinearity is not so

Fig. 7. Eye shape detection using the proposed method. Red points represent the exact eye shapes manually labeled. Green Points represent the eye shapes detected by the proposed method.

Table 1. Comparison between ASM, DT and the proposed method.

Method	Accuracy%	Iteration	Detection Time(msec)
ASM	69	20	85
DT	85	12	240
Prop. Method	90	-	65

high. The structure in Fig. 2 enables the network to learn very high nonlinear functions. This powerful performance is governed by the recurrent connections between the neurons themselves and each other. Figure 5 proves this point by showing the predicted signal of an eye profile. There is a difference in magnitude between the training and predicted signals especially at the actual location of the eye contour. This difference doesn't affect the results because the most significant information is the location of the rapid and sudden change in the predicted signal. As long as the pixel values of an eye profile change smoothly, a sharp change is guaranteed to be generated by the recurrent neural network one time along the eye profile.

On the other hand, the difference between two eye profiles in the same eye image is proportional to θ as highlighted in Fig. 3. Based on learning the raw eye profile in Fig. 5(a), the network was able to predict the location of the eye contour which is 6 degrees distant, Fig. 5(b).

According to the conducted experiments, the recurrent network has provided low performance and almost failed to predict the correct locations of the eye contour when the distance of an eye profile is approximately larger than 10 degrees from the training eye profile. Therefore, the sampling rate S must be selected carefully.

In order to make the network more flexible and robust to deal with different eye profiles of various eye images, we trained the network, with increasing the number of hidden neurons, using a series of eye profiles that extracted from

different eye images at fixed sampling rate of θ as shown in Fig. 6. We guess that this strategy not only enables the network to learn various patterns of the eye profiles. Although, it enables the network to learn the frequency of sharp pulses which form the eye shape in an eye image. Therefore, the performance has surprisingly improved as shown in Fig. 6(b) for predicting the eye shape of different eye image of the learning one in Fig. 6(a).

Figure 7 illustrates the performance of the proposed method to detect the exact eye shapes of the testing images. The accuracy is high and the detected points almost match the manually labeled points.

The proposed method outperforms ASM and DT as highlighted in table 1. ASM and DT use an iterative strategy to detect the eye shape whereas the proposed method needs to achieve constant steps. This reduces significantly the detection time. The accuracy yielded is high because of using the prediction ability of the recurrent neural network as well as its generalization capabilities to deal with new eye profiles.

The authors don't say that the presented structure of the network is the best for modelling and predicting the eye profiles. On the other hand, we tried to use the same structure as a static network without any recurrent connections. The prediction ability was very low. Moreover, recurrent neural networks have widely been employed in neuro-control applications to be learned in real time. Therefore, the weights of the learned recurrent network can be updated online to track the eye shape in real time. This work is left to be investigated in future.

Beside of using the recurrent neural network to detect the eye shape, Log-polar transform has effectively contributed in increasing the accuracy regardless the change of scale, rotation and translation⁽¹¹⁾.

6. Conclusion

We proposed a method to detect the eye shape in grayscale eye images. The eye image is decomposed using Log-polar transform into a series of eye profiles according to a fixed sampling rate. A recurrent neural network is used to model the pixel values in the eye profiles and to predict ahead the locations of the eye contour. The proposed method differs from the other strategies by attaining constant steps without using an iteration technique. The results verified that the proposed method performs high accuracy with noticeably reduction of detection time compared to the conventional methods.

References

- (1) Dmitry Gorodnichy, and Gerhard Roth: "Nose 'Use Your Nose as a Mouse", International Conference on Vision Interface, pp 354-361, 2002
- (2) Jie Zhu, and Jie Yang: "Subpixel Eye Gaze Tracking", International Conference on Automatic Face and Gesture Recognition, pp 131-136, 2002
- (3) Linda Sibert, Robert Jacob, and James Templeman, "Evaluation and Analysis of Eye Gaze Interaction", NRL Report NRL/FR/5513-01-9990, Naval Research Laboratory, Washington, DC, 2001
- (4) Qiang Ji, Zhiwei Zhu, and Peilin Lan: "Real-time Nonintrusive Monitoring and Prediction of Driver Fatigue", IEEE Trans. Vehicles Technology, Vol. 53, No. 4, pp 1052 -1068, 2004
- (5) Takatsugu Hirayama¹, Kenji Mase, and Kazuya Takeda: "Driver Inattention Detection based on Eye Gaze-Road Event Correlation", International Journal of Robotics Research, pp 774-801, 2009
- (6) Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena López: "EOG guidance of a wheelchair using neural networks", International Conference on Pattern Recognition, pp 668-671, 2009
- (7) Timothy F. Cootes , Gareth J. Edwards , Christopher J. Taylor: "Active Shape Models-Their Training and Application Computer", Vision and Image Understanding, Vol. 61, No. 1, pp 38-59, 1995
- (8) Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor: "Active Appearance Models", European Conference on Computer Vision, pp 321-331, 1998
- (9) Alan Yuille, Peter Hallinan, and David Cohen: "Feature Extraction from Faces Using Deformable Templates", International Journal of Computer Vision, Vol. 8, No. 2, pp 99-111, 1992
- (10) Michael Kass, Andrew Witkin, and Demetri Terzopoulos: "Snakes: Active Contour Models", International Journal of Computer Vision, Vol. 1, No. 4, pp 321-331, 1987
- (11) Mohammad Aldibaja, and Shinichi Suzuki: "Eye Shape Detection Based on Properties of Eigeneye Image in Log-Polar Domain", 8th International Symposium on Advanced Science and Technology in Experimental Mechanics, pp 255-261, 2013
- (12) Helder Araujo, and Jorge Dias: "An Introduction to the Log-polar Mapping", IEEE Second workshop on Cybernetic Vision, pp 139-144, 1996
- (13) Nazmul Karim, and Rivera SL: "Comparison of feed-forward and recurrent neural networks for bioprocess state estimation", Computers and Chemical Engineering , Vol. 16, No. 1, pp 369-377, 1992
- (14) Larry Medsker, and Lakhmi Jain: "Recurrent Neural Networks Design and Applications", CRC Press, c2000, CRC International Series on Computational Intelligent
- (15) Ronald Williams, and David Zipser: "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks", Neural Computation, Vol. 1, No. 2, pp 270-280, 1989
- (16) David Rumelhart, Geoffrey Hinton, and Ronald Williams: "Learning Internal Representations by Error Propagation", Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations, Cambridge, MA: IT Press/Bradford Books, 1986
- (17) Hugo Proenca, and Luis Alexandre: "UBIRIS: A noisy Iris Image Database", 13th International Conference Image Analysis, pp 970-977, 2005