A PIXEL-WISE, LEARNING-BASED APPROACH FOR OCCLUSION ESTIMATION OF IRIS IMAGES IN POLAR DOMAIN

Yung-hui Li and Marios Savvides

Carnegie Mellon University, U.S.A.

ABSTRACT

On normalized iris images, there are many kinds of noises, such as eyelids, eyelashes, shadows or specular reflections, that often occlude the true iris texture. If high recognition rate is desired, those occluded areas must be estimated accurately in order for them to be excluded during the matching stage. In this paper, we propose a unified, probabilistic and learning-based approach to estimate all kinds of occlusions within one unified model. Experiments have shown that our method not only estimates occlusion very accurately, but also does it with high speed, which makes it useful for practical iris recognition systems.

Index Terms— iris recognition, biometrics, iris mask, occlusion estimation, Gaussian Mixture Models, FJ-GMM

1. INTRODUCTION

Iris recognition has been acknowledged as one of the most accurate modalities in the biometrics field. Iris recognition is powerful and highly accurate because it captures the randomness in the meshwork of the connective tissue in the iris region. However, during the time iris images are acquired, irises are often partly occluded by other undesired objects, for example, eyelids, eyelashes, shadows or specular reflections, as shown in Fig. 1. In order to achieve a high recognition rate, an iris mask has to be created to indicate which part of the image contains authentic iris texture and which part of the image is contaminated by other artifacts. If iris masks are not accurate, the final recognition rate of the iris recognition system will be poor.

As stated above, there are different kinds of noises, and all of them should be masked out in order to achieve high recognition performance. Because the image properties of different kinds of noises are different, in the past, research has been focused on finding either eyelids or eyelashes. In this paper, we would like to propose a unified approach which can deal with all kinds of occlusions at the same time, by the same model, and with high accuracy. This method will speed up iris occlusion estimation and will be useful in practical systems.

We propose a probabilistic, learning-based approach that can learn the distribution of the true iris texture from examples, and is able to estimate very accurate iris masks for unseen data, all in a short period of time. Experiments have shown the proposed method is powerful and can enhance the overall iris recognition rate.

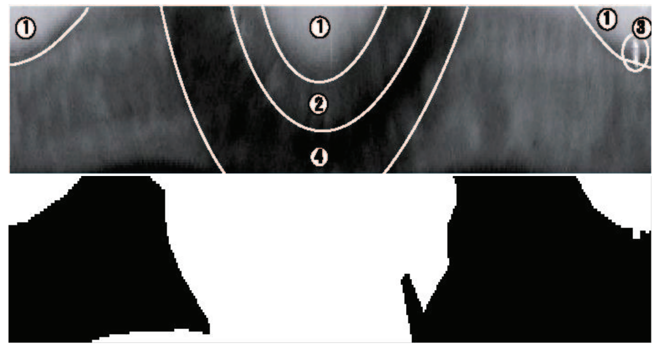

Fig. 1. Normalized iris image and its corresponding mask. Note that there are many types of noises which contaminate the iris texture, including (1) eyelids (2) eyelashes (3) specular reflections and (4) shadows, and all of them have to be masked out.

2. BACKGROUND

In [1], Daugman proposed an optimization scheme for finding the spline parameters that best describe the eyelid boundary. In his later work [2], he proposed a new method, which used active contour (Snake) to find the boundary of the eyelids. On the other hand, in [3], Zhang et al. proposed using Sobel edge filters to detect eyelashes in the polar domain and removing them with median filters.

In [4], Krichen et al. proposed a probabilistic approach for iris quality measure. They compared the performance of the Gaussian Mixture Model with Fourier-based methods, wavelet-based methods and active contour based methods. The iris masks estimated by their method seem to be local patch-based, not pixel-based. In [5], Thornton proposed to use a discriminative learning method based on FLDA to estimate iris masks in the polar domain.

3. PROPOSED METHOD

Let us review the problem of iris occlusion estimation from a machine-learning perspective. The problem of estimating a mask for an iris polar image can be treated as a two-class classification problem. For each pixel on the iris polar image, we should extract robust features that contain discriminative information about whether this pixel belongs to the iris texture or occlusion. We can then train a classifier by using some training data and later use the trained classifier to perform classification for unseen data.

We propose to use Gaussian mixture modeling (GMM) to model the posterior probability distribution of both iris texture and occlusion classes. GMMs have been widely used in all kinds of problems in machine learning and pattern recognition, including speech processing [6], and real-time tracking [7]. The advantage of GMM is its modeling ability. As long as the number of Gaussians is large enough, GMM can virtually model any shape of distribution.

Traditionally one has to determine the number of GMM used and also the initial location of GMM during training process. We propose to use Figueiredo-Jain's extension for GMM (FJ-GMM) to automatically select the best parameters for these two pre-conditions, and avoid GMM to converge to local minima. The math of FJ-GMM is described briefly in the following sub-sections.

3.1. Probability Density Function for GMM

Let us review the basic mathematical foundations for GMM. The Gaussian distribution of a D dimensional random variable X which has a value x is represented by (1)

$$X \sim \mathcal{N}\left(x;\mu,\Sigma\right) = \frac{1}{(2\pi)^{\frac{D}{2}}\left|\Sigma\right|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]} \tag{1}$$

where μ is the mean vector and Σ is the covariance matrix of the Gaussian distributed random variable X.

The probability density function of GMM can be defined as a weighted sum of multiple Gaussian distributions, as shown in (2)

$$p(x;\,\theta) = \sum_{c=1}^{C} \alpha_c \mathcal{N}(x;\,\mu_c;\Sigma_c) \tag{2}$$

where α_c is the priori probability that the random variable $X = x$ is generated by the cth Gaussian mixture. Based on (1)-(2), the probability density function for a Gaussian mixture model can be completely defined by a parameter list as shown in (3)

$$\theta = \{\alpha_1,\,\mu_1,\,\Sigma_1,...,\,\alpha_C,\,\mu_C,\,\Sigma_C,\} \tag{3}$$

Given the observation $X = x$ and the model θ, the likelihood function can be defined as (4)

$$\mathcal{L}(\mathrm{X};\,\theta) = \prod_{n=1}^{N} p(x;\,\theta) \tag{4}$$

It tells the probability that the series of observation $X = x$ is generated by distribution governed by θ. The goal of parameter estimation is to find the optimal parameter $\hat{\theta}$ that maximize the probability:

$$\hat{\theta} = arg\,\max_{\theta} \mathcal{L}(\mathrm{X};\,\theta) \tag{5}$$

Parameter estimation by (5) is called Maximum-Likelihood Estimation (MLE). Sometimes maximum a posteriori (MAP) estimation is used instead of MLE:

$$\hat{\theta}_{MAP} = arg\,\max\left\{\ln\mathcal{L}(\mathrm{X};\,\theta) + \ln\mathcal{L}(\theta)\right\} \tag{6}$$

3.2. Figueiredo–Jain's Extension for GMM Training

Figueiredo and Jain proposed an unsupervised learning method for GMM in [8]. This method can estimate the number of Gaussian mixtures without human intervention, and can avoid the boundary of the parameter space during the converging stage. FJ algorithm uses the idea of Minimum Discriptive Length (MDL) and applies it to mixture model training. It is equivalent to using the objective function in (7)

$$\begin{aligned}\Lambda(\theta,\,\mathrm{X}) \;=\; & \frac{V}{2}\sum_{\alpha_c>0}\ln\left(\frac{N\alpha_c}{12}\right) + \frac{C_{nz}}{2}\ln\frac{N}{12} \\ & + \frac{C_{nz}\,(V+1)}{2} - \ln\mathcal{L}(\mathrm{X},\,\theta)\end{aligned} \tag{7}$$

where N is the number of training points, V is the number of free parameters of the GMM, C_{nz} is the number of Gaussian mixtures that have nonzero weight ($\alpha_c > 0$), θ is defined as in (3), and the last term is log likelihood.

By using (7) as the new objective function, the formula for estimating the prior distribution of the Gaussian mixture in FJ algorithm becomes

$$\alpha_c^{i+1} = \frac{\max\left\{0,\,\left(\sum_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}}{\sum_{j=1}^{C}\max\left\{0,\,\left(\sum_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}} \tag{8}$$

where $w_{n,c}$ is the probability that the n^{th} observation is generated from the c^{th} Gaussian mixtureis, defined as

$$w_{n,c} = \frac{\alpha_c^i p\left(x_n|c;\,\theta^i\right)}{\sum_{j=1}^{C}\alpha_c^i p\left(x_n|c;\,\theta^i\right)} \tag{9}$$

The formula for estimating parameter μ_c and Σ_c is the same as in the traditional EM algorithm:

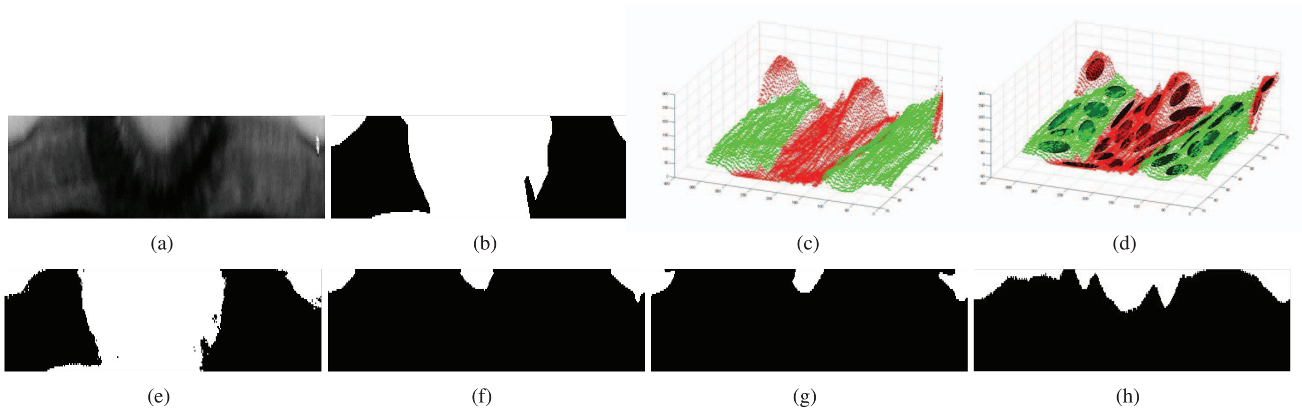

Fig. 2. Visualization of GMM trained on single iris texture. (a) Example image for training; (b) Iris mask of (a), created manually; white region indicates occlusion and black region indicates true iris; (c) Iris image viewed in 3D, where z coordinate is the pixel intensity value; green and red color denotes authentic iris texture and occluded area, respectively; (d) Visualization of GMM trained from (a), plotted as red and green ellipses together with training points; (e)-(h) occlusion mask estimated by GMM, rule-based method, FLDA-based method and Snake-based method, respectively.

$$\mu_c^{i+1} = \frac{\sum_{n=1}^{N} x_n w_{n,c}}{\sum_{n=1}^{N} w_{n,c}} \tag{10}$$

$$\Sigma_c^{i+1} = \frac{\sum_{n=1}^{N} w_{n,c} \left(x_n - \mu_c^{i+1}\right)\left(x_n - \mu_c^{i+1}\right)^T}{\sum_{n=1}^{N} w_{n,c}} \tag{11}$$

We will not repeat too many details about FJ algorithm. Interested readers should refer to [8].

4. EXPERIMENTS AND RESULTS

4.1. GMM Trained on Single Image

We first tried our proposed method on one single iris image to see how the trained GMM fit with the training image. We take image 245241.tiff from ICE2 database to be our training sample. After manually segmenting the iris and performing iris normalization, we get Fig. 2(a). We created manual mask for it, and get Fig. 2(b). We can plot all points in Fig. 2(a) in 3D, using pixel intensity as z coordinate, and get Fig. 2(c). Applying FJ-GMM method to train GMM for those points, we get Fig. 2(d). Finally, using Fig. 2(a) again as test data, an iris mask can be estimated, as shown in Fig. 2(e). The result showed that GMM trained with proposed method can fit both distributions (authentic iris and occlusion) very well and can reconstruct a highly accurate iris mask.

4.2. Automatic Mask Generation on ICE2 Dataset

We also performed a large-scale experiment on automatic mask generation by proposed method, and compared the results with other methods. The database we used is a subset of NIST ICE database, which is ICE2 database, as described in [9]. The iris feature extraction and matching algorithm we used in this experiment was Libor Masek's Matlab implementation of Daugman's algorithm, which is publicly available[10]. For the proposed algorithm, we take one image from each class, manually create their masks, and train two GMMs, one for authentic iris texture and the other for all occlusions (eyelids, eyelashes, shadows and specular reflections). The features of points we used are: x-coordinate, y-coordinate, pixel intensity, mean and standard deviation of pixel intensity on a local 5x5 window. Therefore the number of features per points is five. Since there are 120 classes, and each image is of size 61x360, the total number of points for training is 120x61x360=2,635,200. We test on all other images which are not in the training set.

Method	No Mask	Rule-based	FLDA	Snake	FJ-GMM
Time(sec)	0	0.66	2.21	18.22	0.26

Table 1. The time it takes to estimate iris occlusion for one image (of size 61x360)

The methods we are comparing against include (1) No mask at all; (2) Manually created mask; (3) Rule-based method, similar to [11]; (4) FLDA-based method, as proposed in [5]; (5) Active contour based method. The active contour we used is GVF Snake [12]. The results are plotted in the ROC curve and shown in Fig. 3. We also measured the time each algorithm takes for estimating iris occlusion for one image. They are listed in Table 1.

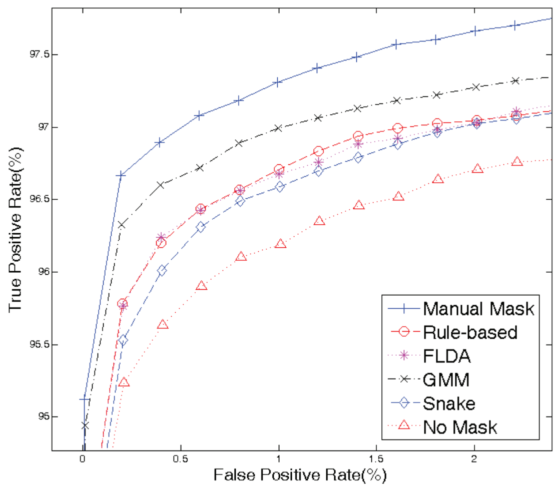

Fig. 3. ROC curves for different masks

5. DISCUSSION AND CONCLUSION

From Fig. 3, we can see that the our proposed method performs best, better than the rule-based and FLDA-based method. Snake performs worst. This may be due to the fact that Snake can only search the eyelids boundary, but there are other artifacts which should be excluded during the matching process, e.g. eyelashes, shadows, or specular reflections. Also, our proposed method takes the least amount of time among all methods, which makes it more useful in practical systems.

We would like to point out that our method, although it seems similar to [4] at first sight, is actually quite different than their approach. There are three major differences. First, their method only models the distribution of true irises, but not on the occlusion; in our proposed method, we model both and the decision is made by comparing the posterior probability of both classes. Second, their feature is naive and simple, the only observation (features of the points) is pixel intensity, while our proposed method use more descriptive features (location, pixel value, local mean and variance) to describe one point. Third, their method is based on local patches, where the whole local patch (size 11x51) has to be classified as a true iris or occlusion; in our proposed method, each pixel is treated as an individual observation and can be classified individually.

Our future work includes using responses from filters as features and optimizing filters for feature extraction in order to get best results. We would also like to try more complex generative models such as Markov Random Field to see if accuracy increases.

6. REFERENCES

[1] J. Daugman, "How iris recognition works," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 21–30, Jan. 2004.

[2] J. Daugman, "New methods in iris recognition," *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, vol. 37, no. 5, pp. 1167–1175, Oct. 2007.

[3] D. Zhang, D.M. Monro, and S. Rakshit, "Eyelash removal method for human iris recognition," *Image Processing, 2006 IEEE International Conference on*, pp. 285–288, Oct. 2006.

[4] E. Krichen, S. Garcia-Salicetti, and B. Dorizzi, "A new probabilistic iris quality measure for comprehensive noise detection," *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pp. 1–6, Sept. 2007.

[5] J. Thornton, "Matching deformed and occluded iris patterns: a probabilistic model based on discriminative cues," *PhD thesis, Carnegie Mellon University*, 2007.

[6] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of speech recognition*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[7] Chris Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *cvpr*, vol. 02, pp. 2246, 1999.

[8] M.A.T. Figueiredo and A.K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[9] "Iris challenge evaluation," *National Institute of Standards and Technology, http://iris.nist.gov/ICE/*, 2006.

[10] Libor Masek and Peter Kovesi, "Matlab source code for a biometric identification system based on iris patterns," *The School of Computer Science and Software Engineering, The University of Western Australia*, 2003.

[11] Jinyu Zuo, N.D. Kalka, and N.A. Schmid, "A robust iris segmentation procedure for unconstrained subject presentation," *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, pp. 1–6, 19 2006-Aug. 21 2006.

[12] Chenyang Xu and J.L. Prince, "Snakes, shapes, and gradient vector flow," *Image Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 359–369, Mar 1998.