Workshop on Biometrics
(in association with CVPR)
Anchorage, Alaska, June 2008

Non-Ideal Iris Segmentation Using Graph Cuts

Shrinivas J. Pundlik* Damon L. Woodard† Stanley T. Birchfield*
*Electrical and Computer Engineering Department
†Image and Video Analysis Lab, School of Computing
Clemson University, Clemson, SC 29634
{spundli, woodard, stb}@clemson.edu

Abstract

A non-ideal iris segmentation approach using graph cuts is presented. Unlike many existing algorithms for iris localization which extensively utilize eye geometry, the proposed approach is predominantly based on image intensities. In a step-wise procedure, first eyelashes are segmented from the input images using image texture, then the iris is segmented using grayscale information, followed by a postprocessing step that utilizes eye geometry to refine the results. A preprocessing step removes specular reflections in the iris, and image gradients in a pixel neighborhood are used to compute texture. The image is modeled as a Markov random field, and a graph cut based energy minimization algorithm [2] is used to separate textured and untextured regions for eyelash segmentation, as well as to segment the pupil, iris, and background using pixel intensity values. The algorithm is automatic, unsupervised, and efficient at producing smooth segmentation regions on many non-ideal iris images. A comparison of the estimated iris region parameters with the ground truth data is provided.

1. Introduction

Automated person identification and verification systems based on human biometrics are becoming increasingly popular and have found wide ranging applications in defense, public, and private sectors. Over the years the iris has emerged as an effective biometric for many such applications due to the availability of efficient algorithms for performing iris recognition tasks. A large number of iris recognition approaches rely on *ideal iris images* for successful recognition, i.e., low noise iris images in which the person is looking straight at the camera. Their performance degrades if the iris undergoes large occlusion, illumination change, or out-of-plane rotation. Iris recognition using such *non-ideal iris images* is still a challenging problem. Figure 1 shows an ideal and several non-ideal iris images.

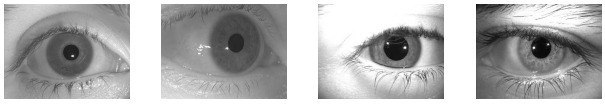

Figure 1. An ideal iris image (left), and iris images of varying quality (right three columns), containing out of plane rotation, illumination effects, and occlusion.

Iris segmentation is an important part of the larger recognition problem, because only once the iris has been localized can the unique signature be extracted. In previous work, geometric approaches have been common. For example, in his pioneering work on iris recognition, Daugman [5, 6] fits a circle to the iris and parabolic curves above and below the iris to account for eyelids and eyelashes. Similarly, geometric cues such as pupil location or eyelid location have been used for iris localization [8], while stretching and contraction properties of the pupil and iris have also been used [10]. Another important approach has been to detect the eyelashes in order to determine iris occlusion. To this end, Ma *et al*. [13] use Fourier transforms to determine whether the iris is being occluded by the eyelashes; the unique spectrum associated with eyelashes are used to reject images in which significant iris occlusion occurs. Other approaches for eyelash segmentation involve the use of image intensity differences between the eyelash and iris regions [12, 11], gray level co-occurrence matrices [1], and the use of multiple eyelash models [16].

These attempts at iris segmentation are limited to ideal iris images, assuming that the shape of the iris can be modeled as a circle. Such a simplifying assumption limits the range of input images that can be successfully used for recognition. By relying on geometry, these techniques are sensitive to noise in the image. Some more recent approaches to handle non-ideal iris images rely upon active contour models [7] or geodesic active contours [14] for iris segmentation. Building upon this work, we propose in this paper an algorithm for eyelash and iris segmentation that uses image intensity information directly instead of relying

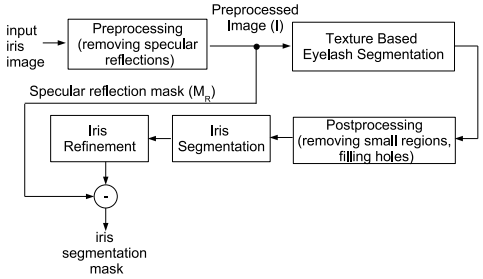

Figure 2. Overview of the proposed approach.

on intensity gradients. Our approach models the eye as a Markov random field and uses graph cuts to minimize an objective function that enforces spatial continuity in the region labels found. Four labels are assigned: iris, pupil, eyelashes, and background, in addition to specular reflections. By automatically choosing the expected graylevel values via histogramming, the algorithm adapts to variations in images, enabling it to handle non-ideal iris images containing out-of-plane rotation, extensive iris occlusion by eyelashes and eyelids, and various illumination effects.

An overview of our approach is presented in Figure 2. The first step is a simple preprocessing procedure applied to the input images to deal with specular reflections which may cause errors in segmentation. In the second step we perform texture computation for eyelash segmentation by measuring the amount of intensity variations in the neighborhood of a pixel and generating a probability map in which each pixel is assigned a probability of belonging to a highly textured region. This pixel probability map is fed to an energy minimization procedure that uses graph cuts to produce a binary segmentation of the image separating the eyelash and non-eyelash pixels. A simple postprocessing step applies morphological operations to refine the eyelash segmentation results. The next step is to segment iris images based on grayscale intensity. The iris refinement step involves fitting ellipses to the segmented iris regions for parameter estimation. The final step is to combine the iris region mask and the specular reflection mask to output usable iris regions. These steps are described in more detail in the following sections.

2. Removing Specular Reflections

Specular reflections are a major cause of errors in iris recognition systems because of the fact that the affected iris pixels cannot be used for recognition . In this case, these bright spots (see Figure 3) are a cause of segmentation error as high texture values are assigned to the pixels surrounding these points which are in turn segmented as eyelashes. We adopt a straightforward preprocessing procedure to remove the specular reflections from the input iris images. Let R be the raw input image. The output of this preprocessing

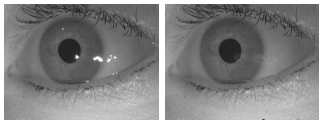

Figure 3. Removing specular reflection in iris images. LEFT: Input image. RIGHT: Preprocessed image with specular reflections removed.

step is the preprocessed iris image I with the reflections removed and a binary mask M_R corresponding to the pixels removed from R. We maintain a list of all the pixel locations in the input image with grayscale intensity higher that a preset threshold value along with their immediate neighbors. The values of the pixel locations in the list are set to zero, i.e., these pixels are unpainted. The list is sorted according to the number of painted neighbors each pixel has. Starting from the first element in the list, grayscale values are linearly interpolated until all the unpainted pixels are assigned a valid gray value. Results of the specular reflection removal algorithm are shown in Figure 3. It should be noted that the *painted* pixels obtained by the above algorithm cannot be used for iris recognition and are discarded or masked while constructing the iris signatures.

3. Segmentation of Eyelashes

3.1. Texture Computation

Let I be an image with N pixels, and let I_x and I_y denote the derivatives of the image in the x and y directions, respectively. For each image pixel n, texture is computed using the gradient covariance matrix, which captures the intensity variation in the different directions [15]:

$$G(n) = \sum_{n' \in \mathcal{N}_g(n)} \begin{bmatrix} I_x^2(n') & I_x(n')I_y(n') \\ I_x(n')I_y(n') & I_y^2(n') \end{bmatrix}, \quad (1)$$

where $\mathcal{N}_g(n)$ is the local neighborhood around the pixel. If both the eigenvalues of $G(n)$ are large, then the pixel n has large intensity variations in orthogonal directions. This is usually known as a point feature and is indicative of a high amount of texture in its immediate neighborhood. Letting e_1 and e_2 be the two eigenvalues of $G(n)$, we detect points for which $h(n) = \min\{e_1, e_2\} > \tau$, where τ is a threshold. The value $h(n)$ is indicative of the quality of the feature. Depending upon the value of τ, we can adjust the quality and hence the number of such points detected in any image.

Let f_i be the i^{th} point feature detected in the image with corresponding weight $h(f_i) > \tau$, $i = 1, \ldots, M$. Here $M \ll N$, i.e., the number of point features detected is much less than the number of image pixels. We need a dense map that assigns a probability value to each pixel in the input image. To accomplish this, we compute an oriented histogram of point features weighted by their values in a region around

a pixel in an image. This spatial histogram is defined by two concentric circles of radii r_1 and r_2 centered around a pixel n. The inner and outer circular regions are represented by H_n and $\overline{H}_n$, respectively. These regions are divided into K bins, each spanning $(360/K)$ degrees and carrying an equal weight of ω_b. The bin values of this 2D oriented histogram are further multiplied by the weights associated with the circular region of which it is a part, i.e., bins in the inner circle are weighted by ω_{r_1} while the outer ones are weighted by ω_{r_2}. The feature point score at a pixel n is obtained from the normalized sum of all the bins at a point:

$$\mathcal{P}_f(n) = \frac{1}{K}\sum_{k=1}^{K}\omega_b \left\{ \sum_{f\in H_n(k)} \omega_{r_1} h(f) + \sum_{f\in \overline{H}_n(k)} \omega_{r_2} h(f) \right\}, \tag{2}$$

where $H_n(k)$ and $\overline{H}_n(k)$ are the set of features contributing to the k^{th} bins of the two histograms.

The feature point score alone cannot give a substantive measure of texture in an image because the feature points represent locations where image intensity changes occur in both x and y directions. To effectively compute the texture around a point, we have to account for all the neighboring points with gradient changes in a single direction. To address this problem, we sum the gradient magnitudes in the neighborhood of a pixel in a manner similar to the one described above in the case of finding the feature point score in Equation (2). The score due to gradients is given by

$$\mathcal{P}_g(n) = \frac{1}{K}\sum_{k=1}^{K}\omega_b \left\{ \sum_{j\in \mathcal{R}(k)} \omega_{r_1} g(j) + \sum_{j\in \overline{\mathcal{R}}(k)} \omega_{r_2} g(j) \right\}, \tag{3}$$

where

$$g(j) = \sqrt{I_x^2(j) + I_y^2(j)}$$

is the gradient magnitude sum in the j^{th} pixel in a histogram, and $\mathcal{R}(k)$ and $\overline{\mathcal{R}}(k)$ are the image regions specified by the k^{th} bins of the two histograms.

The total score for a pixel is the sum of the feature point score and the gradient score:

$$\mathcal{P}(n) = \mathcal{P}_f(n) + \mathcal{P}_g(n). \tag{4}$$

We compute the total score for each pixel and normalize the values to obtain a probability map that assigns the probability of each pixel having high texture in its neighborhood. Figure 4 shows the various texture measures and the texture probability map obtained for an iris image.

3.2. Image Bipartitioning using Graph Cuts

Once the texture probability map is obtained for an input image, it is desirable that the segmentation produces smooth

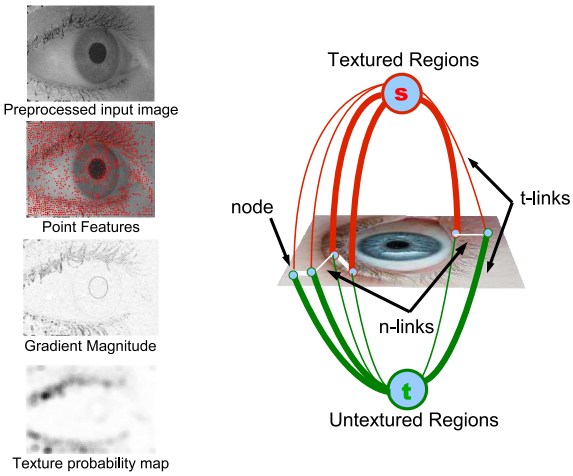

Figure 4. Eyelash segmentation details. LEFT: Steps involved in the texture computation. RIGHT: Binary graph cuts on an image. For clarity, only a few nodes and corresponding links are shown. Thicker links denote greater affinity between the corresponding nodes or terminals (i.e., t-links between terminals and nodes and n-links between two nodes).

regions as an output. This problem can be considered as a binary labeling problem. Our goal is to assign a label $l \in \{0, 1\}$ to each pixel in the image based on the probability map $\mathcal{P}$. Let $\psi : \mathbf{x} \rightarrow l$ be a function that maps an image pixel $\mathbf{x}$ to a label l. If $D_n(l_n)$ represents the energy associated with assigning label l_n to the n^{th} pixel, then the energy term to be minimized is given by

$$E(\psi) = E_S(\psi) + \lambda E_D(\psi), \tag{5}$$

where

$$E_S(\psi) = \sum_{n=1}^{N}\sum_{m\in\mathcal{N}_s(n)} S_{n,m}(l_n, l_m) \tag{6}$$

$$E_D(\psi) = \sum_{n=1}^{N} D_n(l_n). \tag{7}$$

In these equations, $E_s(\psi)$ is the smoothness energy term that enforces spatial continuity in the regions, , N is the number of pixels in the image, $\mathcal{N}_s(n)$ is the neighborhood of the n^{th} pixel, and λ is the regularization parameter. The data penalty term, derived from $\mathcal{P}$, is given by:

$$D_n(l_n) = \exp\{\rho(l_n - \mathcal{P}(n))\},$$

where

$$\rho = \begin{cases} 1 & \text{if } l_n = 1 \\ -1 & \text{if } l_n = 0 \end{cases}.$$

The smoothness term is given by:

$$S_{m,n}(l_m, l_n) = [1 - \delta(m, n)] \exp\{-\|I(m) - I(n)\|^2\},$$

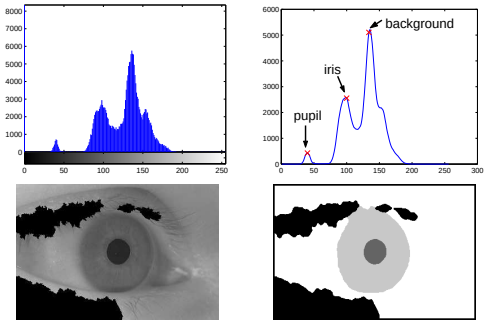

Figure 5. Iris segmentation. TOP: Grayscale histogram of a typical iris image and a smoothed version on the right with peak detection. BOTTOM: Iris image for which the histogram is computed and the corresponding segmentation.

where $\delta(m, n) = 1$ when $m = n$, or 0 otherwise. $I(m)$ and $I(n)$ are image intensities of m^{th} and n^{th} pixels, respectively.

The energy term in Equation (5) is minimized by a graph cut algorithm [2]. The image can be considered as a weighted graph $G(\mathcal{V}, \mathcal{E})$, where the vertices $\mathcal{V}$ are the pixels, and the edges $\mathcal{E}$ are the links between neighboring pixels. For a binary graph cut problem, two additional nodes known as source and sink terminals are added to the graph. The terminals correspond to the labels being assigned to the nodes, i.e., pixels of the image. In this case, the source terminal corresponds to the high-texture label, while the sink terminal is associated with the low-texture label. A cut $\mathcal{C}$ is a set of edges that separates the source and sink terminals such that no subsets of the edges themselves separate the two terminals. The sum of the weights of the edges in the cut is the capacity of the cut. The goal is to find the minimum cut, i.e., the cut for which the sum of the edge weights in the cut is minimum. Figure 4 is a representative diagram showing the process of partitioning the input image.

4. Iris Segmentation

Iris segmentation is based upon the same energy minimization approach described in the previous section, except that it involves more than two labels. In fact, for a typical image, four labels are considered: eyelash, pupil, iris, and background (i.e., the rest of the eye). Since the eyelash segmentation already provides us with a binary labeling that separates the eyelash pixels, our problem is reduced to that of assigning labels to the remaining pixels in the image. Although this is an NP-hard problem, the solution provided by the α-β *swap* graph-cut algorithm [3] is in practice a close approximation to the global minimum. The algorithm works by initially assigning random labels to the pixels. Then for all possible pairs of labels, the pixels assigned to those labels are allowed to swap their label in order to minimize the energy of Equation (5). The new labeling is retained only if the energy is minimized, and this procedure is repeated until the overall energy is not further minimized. Convergence is usually obtained in a few (about 3–4) iterations. Grayscale intensities of the pixels are used to compute the data energy term of Equation (7). Figure 5 shows the grayscale histogram of a typical image of an eye. The three peaks in the histogram correspond to the grayscale intensities of the pupil, iris, and background. The desired grayscale values for the pupil, iris, and background regions are obtained via a simple histogram peak detecting algorithm, where we assume that the first local maximum corresponds to the pupil region, the second to the iris, and so on. Figure 5 shows the iris segmentation obtained using this approach.

The quality of iris segmentation depends on the nature of the image and is highly susceptible to noise and illumination effects in the input images. To overcome these problems, we use a priori information regarding the eye geometry for refining the segmentation of the iris region. Specifically, we assume the iris can be approximated by an ellipse centered on the pupil and aligned with the image axes. Even if these assumptions are not valid for some images, they serve as a good starting point for estimating the iris region. The previous segmentation step provides us with a location of the pupil center. In our experiments, we observed that the pupil is accurately segmented in almost all cases even if the overall image quality is poor. However, in certain cases, other dark regions are mistakenly labeled as pupil. These mistakes are easily corrected by enforcing a maximum eccentricity on the dark region to distinguish the true pupil from these distracting pixels.

In order to find the best fitting ellipse to the segmented iris region, points near the iris boundary must be reliably located considering the possibilities that the segmented iris region may not have a elliptical shape, and that the iris may be occluded partly by the eyelashes (on the top or bottom or both). In other words, even if we know the approximate location of the center of the iris (i.e., the pupil center), its exact extent in both the x and y directions cannot be naively ascertained using the segmented iris regions. For a reliable initial estimate of iris boundary points, we extend rays from the pupil center in all directions (360°) with one degree increments and find those locations where the lines transition from an iris region to the background region (see Figure 6). Because all these lines extending out from a center point may not lead to an iris boundary point, only a subset of the 360 points is obtained. To increase the number of points (and hence increase the reliability of the ellipse fitting procedure), we utilize the inherent symmetry of the iris region. For each ellipse point, a new point is generated about the vertical symmetry line passing through the center of the iris, if a point does not already exist for that direction. In addition, points whose distance from the pupil center exceeds

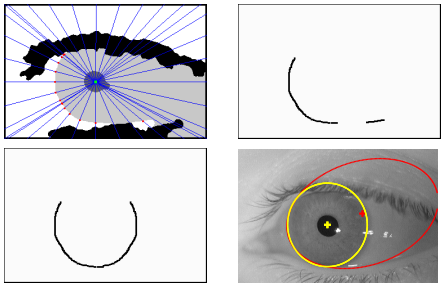

Figure 6. Refining the iris segmentation. TOP LEFT: Iris segmentation image with pupil center overlaid (green dot). The lines originating from the center point in 360° of the center point intersect with the iris boundary at points shown in red. For clarity only a subset of lines and corresponding points are shown. TOP RIGHT: Potential iris boundary points. Due to erroneous segmentation, the full set of points is not obtained. BOTTOM LEFT: Increasing the iris boundary points using the pupil center and the inherent symmetry in the iris regions. BOTTOM RIGHT: Ellipse fitting to the potential iris boundary points leads to an erroneous result (red ellipse), while fitting to the increased boundary points leads to the correct result (yellow ellipse).

1.5 times the distance of the closest point to the pupil center are rejected. This yields a substantial set of points to which an ellipse is fit using the least squares method proposed by Fitzgibbon *et al*. [9]. Figure 6 summarizes this process and shows the results of our ellipse fitting algorithm.

5. Experimental Results

We tested our approach on various non-ideal iris images captured using a near infrared camera. Figure 7 shows the results of our approach on some sample images obtained from the West Virginia University (WVU) Non-Ideal Iris database, [4] (a sample of images can be found online[1]). It can be seen that each step in our approach aids the next one. For example, eyelash segmentation helps in iris segmentation by removing the eyelashes which may cause errors in iris segmentation. To perform eyelash segmentation we used 8-bin histograms for computing feature points and gradient scores $(K = 8)$. The bin weight, ω_b, is set at 0.125 while $w_{r_1} = 1$, $w_{r_2} = 0.75$, and $\tau = 50$. It can be seen that despite using a simple texture measure, the algorithm is able to accurately segment regions. The iris segmentation step, in turn, helps the iris refinement step, and the preprocessing step to remove specular reflections is also helpful in iris segmentation and building a mask of usable iris regions.

To quantitatively evaluate our results we compared our iris localization results with direct ground truth. We used 60 iris images (40 with out-of-plane rotation) from the WVU Non-Ideal Iris image database for iris localization and verification. We manually marked the iris regions in the input images and obtained the ground truth parameters such as the location of the center of the iris and the x and y radius values. We also obtained a mask of the usable iris regions (without specular reflections) from the original image. The parameters of our estimated iris region were compared with ground truth in terms of the iris center location, x and y radius, and the number of pixels in agreement with the iris label. Table 1 shows that the average error in the estimation of iris region parameters as compared to the ground truth is small, indicating accurate segmentation and localization.

[1] `http://www.csee.wvu.edu/~xinl/demo/nonideal_iris.html`

Iris Parameter	Average Error (in pixels)	Standard Deviation (in pixels)
Center (x)	1.9	2.2
Center (y)	2.7	2.5
Radius (x)	3.4	5.2
Radius (y)	3.9	4.0
Pixel labels	5.9%	7.2%

Table 1. Comparison of estimated iris region parameters with the ground truth data for 60 images from the WVU Non-Ideal Iris database.

6. Conclusion

This paper presents a novel approach for non-ideal iris localization using graph cuts. Key components of the approach include a novel texture-based eyelash segmentation technique which helps in accurate iris segmentation and localization. Unlike many of the existing approaches which use extensive eye geometry heuristics, our approach uses eye geometry for refining the results of iris segmentation and is not overly constrained by the location and size of the iris in the input images. Since we explicitly segment the eyelashes from the input images, we can account for the occlusion of the iris by the eyelids and eyelashes. An added advantage of this algorithm is that it can handle specular reflections that affect iris recognition procedures by removing them from the detected iris regions.

Many improvements can be made to the existing approach at each stage of its operation. The texture measure used by the current algorithm can be modified by including gradient orientation cues to improve the accuracy of eyelash segmentation. The current iris segmentation is somewhat limited as it relies on histogram peaks of the images to assign labels; therefore, multi-modal distributions of intensities in any of the regions can lead to errors. This can be improved by using an approach that uses both intensity distributions and intensity edges to compute the objective function. Another important improvement to be made is to reduce the overall computation time of the algorithm. Currently, the amount of time required for the eyelash segmentation is about two seconds for eyelash segmentation and an additional three seconds for iris localization using

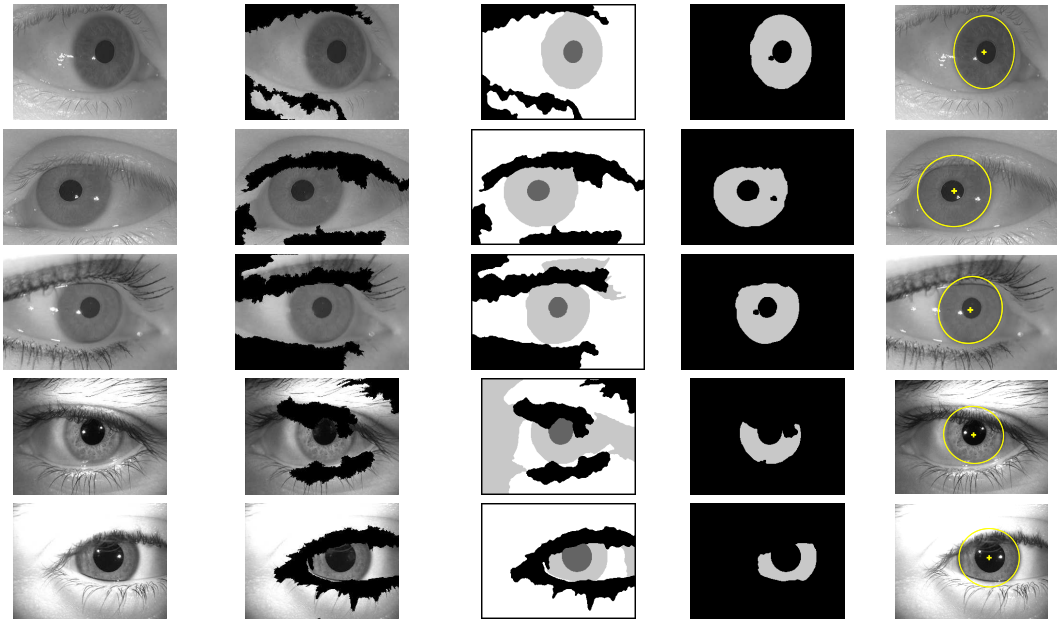

Figure 7. Experimental results of the proposed approach on a sample of iris images from the WVU Non-Ideal Iris image database.

unoptimized C++ code. Finally, our future work involves using the proposed iris segmentation algorithm to perform iris recognition on various available databases to evaluate its performance.

References

[1] A. Bachoo and J. Tapamo. Texture detection for segmentation of iris images. *SAICSIT*, pages 236–243, 2005.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9), 2004.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[4] S. Crihalmeanu, A. Ross, S. Schuckers, and L. Hornak. A protocol for multibiometric data acquisition, storage and dissemination. In *Technical Report, WVU, Lane Department of Computer Science and Electrical Engineering*, 2007.

[5] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[6] J. Daugman. Statistical richness of visual phase information: update on recognizing persons by iris patterns. *International Journal of Computer Vision*, 45(1):23–38, 2001.

[7] J. Daugman. New methods in iris recognition. *IEEE Tran. on System, Man and Cybernetics-Part B: Cybernetics*, 37(5):1167–1175, 2007.

[8] X. Feng, C. Fang, X. Ding, and Y. Wu. Iris localization with dual coarse to fine strategy. In *Proceedings of the IAPR International Conference on Pattern Recognition*, 2006.

[9] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least-squares fitting of ellipses. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999.

[10] Z. He, T. Tan, and Z. Sun. Iris localization via pulling and pushing. In *Proceedings of the IAPR International Conference on Pattern Recognition*, 2006.

[11] B. Kang and K. Park. A robust eyelash detection based on iris focus assessment. *Pattern Recognition Letters*, 28:1630–1639, 2007.

[12] W. Kong and D. Zhang. Detecting the eyelash and reflection for accurate iris segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, pages 1025–1034, 2003.

[13] L. Ma, T. Tan, Y. Wang, and D. Zhang. Personal identification based on iris texture analysis. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 25(12):1519–1533, 2003.

[14] A. Ross and S. Shah. Segmenting non-ideal irises using geodesic active contours. In *Proceedings of Biometrics Symposium*, 2006.

[15] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[16] G. Xu, Z. Zhang, and Y. Ma. Improving the performance of iris recognition systems using eyelids and eyelash detection and iris image enhancement. In *Proceedings of 5th Intl. Conf. on Cognitive Informatics*, 2006.