COMPARISON OF FEATURES RESPONSE IN TEXTURE-BASED IRIS SEGMENTATION

Asheer Bachoo* **and Jules-Raymond Tapamo****

* *Optronic Sensor Systems, Defence Peace Safety and Security, Council of Scientific and Industrial Research, Pretoria, 0001, South Africa*
Email: abachoo@csir.co.za

** *School of Computer Science, University of Kwazulu-Natal, Durban, 4000, South Africa*
Email: tapamoj@ukzn.ac.za

Abstract: Identification of individuals using iris recognition is an emerging technology. Segmentation of the iris texture from an acquired digital image of the eye is not always accurate - the image contains noise elements such as skin, reflection and eyelashes that corrupt the iris region of interest. An accurate segmentation algorithm must localize and remove these noise components. Texture features are considered in this paper for describing iris and non-iris regions. These regions are classified using the Fisher linear discriminant and the iris region of interest is extracted. Four texture description methods are compared for segmenting iris texture using a region based pattern classification approach: Grey Level Co-occurrence Matrix (GLCM), Discrete Wavelet Transform (DWT), Gabor Filters (GABOR) and Markov Random Fields (MRF). These techniques are evaluated according to their true and false classifications for iris and non-iris pixels.

Keywords: iris, texture features, segmentation, pattern classification

1. INTRODUCTION

The iris [1] begins its formation in the 3rd month of gestation. It is a multilayered texture and an assortment of variations are possible. They include contractile lines related to the state of the pupil, irregular atrophy of the border layer (crypts), small elevations of the border layer (naevi), collections of chromatophores (freckles) and color variation.
Segmentation of an iris image is a classical image processing problem. The following occurrences are possible in the acquired iris image:

- Bright lighting can cause specular reflection off the eye, which makes the processing stage almost impossible at times.
- Poor lighting can hide the textural details and introduce an uneven illumination component.
- Atmospheric conditions and human emotion affect the state and size of the pupil. This causes the iris region to vary in size.
- The iris may be partially hidden. This can be caused by eyelashes and eyelids.

The focus of this research paper is to improve the iris segmentation process by considering the above image components as a normal occurrence in an iris image. The iris region that must be extracted has texture properties that are *different* from those of pixels of eyelashes, reflection, pupil and eyelids. This provides the basis for a texture feature extraction and pattern classification approach for segmenting the different components from the iris image. We compare four texture description methods for iris segmentation: grey level co-occurrence matrix, discrete wavelet transform, Gabor filters and Markov random fields.

Several approaches are documented that account for poor or corrupted iris texture. Eyelash and reflection detection has been proposed by Kong and Zhang [2]. They divide the eyelash problem into two possibilities: separable eyelashes and multiple eyelashes. Separable eyelashes are treated as edges whereas multiple eyelashes are modelled using an intensity variation model - eyelashes overlapping in a small area have a low intensity variation. If the variance of intensity in the area is below a threshold, the center of the window is labelled an eyelash pixel. Reflections are defined as strong and weak. A pixel of strong reflection has an intensity value greater than a specified threshold; a pixel of weak reflection is a transition region between strong reflection and the iris. The approach described above is effective. However, there is a use of thresholding for determining eyelash pixels and the threshold is not automatically determined. If the region being thresholded does not have two distinct grey level distributions - corresponding to iris and eyelash pixels - the algorithm may fail. Incorrect threshold selection will also affect the result negatively.

In the work of Huang *et. al.* [3], occlusion by eyelids, eyelashes and specular reflection is considered. Edge data is extracted using phase congruency information and this is then used to determine the most likely boundaries of noise and/or occlusion. The frequency distribution of

iris images has been analyzed to determine occlusions by eyelids and eyelashes [4]. Frequencies in the Fourier domain provide an insight into the content of the iris region - frequencies outside a specified range signify occlusion by eyelids and/or eyelashes. This enables the system to accept or reject an image for processing. Although effective, the technique does not provide a solution to removing the useless regions. In the next section, the methods implemented are discussed.

2. METHODS

The proposed algorithm for segmenting iris images has several important steps. Firstly, an iris image is provided as input to the system and the iris boundaries are located. Therafter, the located iris region is normalized using a transform that makes it invariant to size. The normalized image also has its contrast enhanced. Texture features are then computed for the normalized iris region. These features are either used to train a classifier or segment the normalized iris region i.e. remove noise components. The images processed during this investigation were taken from the *CASIA Iris Database* and are 8-bit grey scale [5]. This database is available from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences.

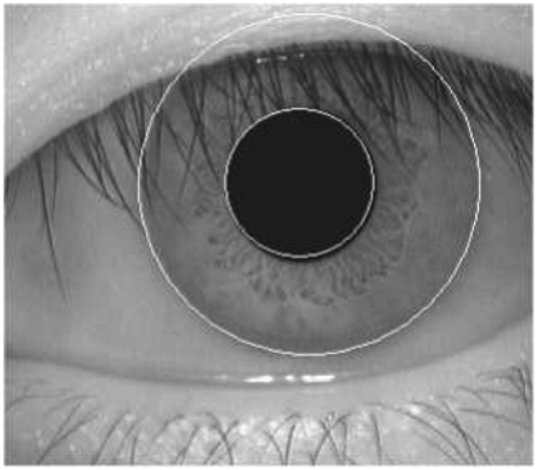
2.1 Iris boundary localization

The inner and outer iris contours are located by modelling them as circles and the eyelids are modelled using parabolic arcs. This proceeds as follows:

1. The input image is globally thresholded using an empirically determined grey scale value of 70. This step segments most of the pupil and provides a binary image. Canny edge detection is applied on the binarized image and an edge map is generated [6]. The pupil is then localized by an application of the Hough transform on the edge map for circle detection [7].

2. The input image is processed in a second pass during which greyscale opening and closing is performed. This processing has the effect of reducing the noise presented by eyelashes and reflection. Thereafter, the image contrast is enhanced by linearly stretching its histogram. An edge map is then generated from the enhanced image using the Canny edge detector and this data is fed to the Hough transform described above. The position of the localized pupil is used as a reference point for finding the center of the circle that lies on the outer iris boundary.

3. The edge map generated in the above step is then processed by a least squares algorithm that fits parabolic arcs to the eyelids. The RANSAC algorithm is incorporated into this process in order to minimize the effect of outliers [8]. The circle parameters are used to reduce the search space for eyelid points and to help differentiate between upper and lower eyelids.



(a) Input iris image with located boundaries

(b) Normalized iris region with eyelids masked out

Figure 1: Iris image pre-processing

The parabolic boundaries computed in the above steps are used to mask out the eyelid pixels when texture analysis is performed.

2.2 Iris normalization

Daugman's rubber sheet model is used to transform the extracted iris region to a dimensionless polar coordinate system [9]. The normalized image is a rectangular grid of size 544×96 pixels. This ensures continuity of texture pixels and also allows fast execution of the discrete wavelet transform. Eyelid regions are then masked out using the parabolas fitted to the eyelid boundaries. Uneven illumination in the iris image causes discontinuities that break up objects when segmentation is performed. It may also degrade object and texture details, making different objects appear the same. The opposite can also occur - similar objects can appear very different. As a result, the multi-scale Retinex [10] is used for illumination flattening before contrast enhancement is performed by histogram adjustment [11]. Figure 1(a) shows an input iris image with its boundaries located. Figure 1(b) is the normalized iris region with the eyelids removed.

2.3 Texture segmentation

In real world problems, images do not demonstrate uniform intensities - they contain variations in tonal content. This represents the textures in the image and their parameters are estimated during the processing stage.

These parameters give texture primitives varying degrees of fineness, coarseness and periodicity of patterns. There is no universal definition of texture since definitions are related to the method of analysis [12]. Texture can be described in terms of texture strength. Strong texture has well defined primitives with a regular structure - elements and spatial relations are easily determined. Weak textures have primitives and spatial relations that are difficult to define - they are referred to as random [13]. It can be regarded as a grouping of similarity in the image [14]; a repetition of basic structural patterns [15]; or as intensity variations that follow a particular periodicity [16]. Once localization and normalization of an iris region has been performed, texture features are computed for the region pixels. These features are filtered using local averaging. This establishes a set of region properties for each pixel. Thereafter, pixels with similar properties are merged into regions and these regions are classified using pattern classification. A brief description of the implemented texture analysis techniques follows.

Grey Level Co-occurrence Matrix (GLCM): Discriminating features for texture separation can be computed using the statistical approach of grey level co-occurrence matrices [17,18]. A GLCM is a matrix of second order statistics that represents pixel configurations as probabilities of pairwise grey level occurrences. These pairwise occurrences must satisfy a particular relationship in order to contribute to the probability matrix. The pixel-pair relationship denotes a spatial dependency for a particular texture. These dependencies are observed in the GLCM, from which a number of features can be computed. This approach has been found to be very popular and effective [19–22].
Considering an $N \times N$ window in a textured image, the configuration of gray-levels can be represented by the matrix $P_{\theta,d}(i,j)$ describing how frequently two pixels with gray-levels i, j appear in the window separated by a distance d in the direction θ [13]. From $P_{\theta,d}$ several Haralick texture features can be extracted, among which some have been used for our experiments [17, 18]. They are:

$$\text{CONT} = \sum_{i,j=1}^{N} (i-j)^2 p_{\theta,d}(i,j) \tag{1}$$

$$\text{ENTR} = -\sum_{i,j=1}^{N} p_{\theta,d}(i,j) \log p_{\theta,d}(i,j) \tag{2}$$

$$\text{MEAN}_1 = \mu = \mu_x = \sum_{i,j=1}^{N} i P_x(i,j) \tag{3}$$

$$\text{SDEV} = \sqrt{\sum_{i,j=1}^{N} P_x(i,j)(i-\mu)^2} \tag{4}$$

where

$$P_x(i) = \sum_{j=1}^{N} p_{\theta,d}(i,j) \tag{5}$$

The features above are the contrast (CONT), entropy (ENTR), mean (MEAN_1) and standard deviation (SDEV) of the GLCM. These features were selected by considering the guidelines presented in [23]. For computational efficiency and improved co-occurrence relations of pixels, we reduce the number of grey levels in the input image to 32. This is done using a straight forward linear scaling. We use $\theta \in \{0, 45, 90, 135\}$, $d \in \{1\}$ and a 9×9 window for computing GLCMs. These parameters produce 16 features for each pixel in the input image.

Discrete Wavelet Transform (DWT): Wavelets are multi-scale transforms that can be used to characterize texture [24]. They provide information about the frequency content of an image. In our context, the DWT applies Haar wavelets to decompose images into Low-Low (LL), Low-High (LH), High-Low (HL) and High-High (HH) frequency components.
The DWT is applied to a 2D digital image $f(x,y)$ by filtering across the rows and then the columns of the result [25, 26]. This is followed by downsampling by a factor 2 to achieve the effect of scaling. The filtering process produces detail (d_j) and approximation (a_j) coefficients at scale j. Processing for the next level ($j+1$) consists of using the a_j as the input. These are the high and low frequency components respectively. The filtering and downsampling operations can be represented using 2×2 kernels which correspond to different frequency components e.g. High-High or Low-High. Convolving every 2×2 non-overlapping block of pixels produces a result equivalent to applying a set of filters across the rows and columns and then downsampling. Kernels for a generalized Haar transform are shown in Figure 2.
Features computed are:

$$\text{MEAN}_2 = \frac{1}{N^2} \sum_{i=1}^{N} d(i)_j \tag{6}$$

$$\text{AAD}_1 = \frac{1}{N^2} \sum_{i=1} \mid d(i)_j - \text{MEAN}_2 \mid \tag{7}$$

$$\text{ENER}_1 = \frac{1}{N^2} \sum_{i=1}^{N} [d(i)_j]^2 \tag{8}$$

where $d(i)$ is a detail coefficient at level j and N is the number of coefficients at level j. ENER measures the signal energy - high energies correspond to high frequency variations. It measures variations in texture patterns at different resolutions. AAD_1 measures the dispersion of coefficients around the mean. MEAN_2 is the common statistical property of a set of samples.
A generalized Haar algorithm decomposes the image, extracting detail and approximation coefficients [24,27]. To perform feature extraction, a 8×8 window is centered at each pixel and 2 passes of the Haar algorithm is performed on windowed pixels together with computation of the statistical information. At each scale, 4 sub-images are produced from the detail and approximation coefficients -

1	1	-1	-1	-1	1	1	-1
1	1	1	1	-1	1	-1	1

Figure 2: Haar kernels operators LL, LH, HL, HH

LL, LH, HL and HH. Features are computed for LL, LH and HL. For LL, the MEAN_2 and AAD_1 are computed. For LH and HL, ENER_1 is computed. Thus, for each pass, 4 features are computed. In addition, the MEAN_2 and AAD_1 are computed for the original image. This provides a total of 10 features for the DWT transform of a texture.

Gabor Filters (GABOR): A popular method for texture feature extraction is the multi-channel filtering approach using Gabor filters [28–30]. This technique is able to exhibit some characteristics of the human visual system. It uses a multi-resolution system to extract information that describes different characteristics of an image. An image is convolved with a set of filters and the outputs are processed to establish a set of texture properties for a pixel.
The Gabor function implemented has the following form:

$$g(x,y) = exp\{-\frac{1}{2}(\frac{x^2}{{\sigma_x}^2} + \frac{y^2}{{\sigma_y}^2})\}cos(\frac{2\pi}{\lambda} + \phi) \quad (9)$$

where:

$$\sigma_x = \frac{\sqrt{ln2}(2^{B_f}+1)}{\frac{\sqrt{2}\pi}{\lambda}(2^{B_f}-1)} \quad (10)$$

$$\sigma_y = \frac{\sqrt{ln2}}{\frac{\sqrt{2}\pi}{\lambda}tan(\frac{B_\theta}{2})} \quad (11)$$

Frequency of the sinusoids is denoted $\frac{1}{\lambda}$. The spread of the Gaussian in the x and y directions is controlled by σ_x and σ_y respectively. The frequency bandwidth of the filter is represented by B_f, the angular bandwidth by B_θ and the phase is ϕ. Spatial rotation of the filter by θ degrees can be achieved using:

$$x^{'} = x\cos\theta + y\sin\theta \quad (12)$$

$$y^{'} = -x\sin\theta + y\cos\theta \quad (13)$$

and then substituting these new values for x and y.
The Gabor kernels require a number of parameters for computation. The value for B_f is set to 1 (octave) and B_θ is 30 degrees [31]. Given a set of λs, the corresponding σs can be determined from the σ:λ ratio. The application dependent parameters are λ, θ, ϕ and γ. γ is set to 0.5 and $\theta \in \{0°, 30°, 60°, 90°, 120°, 150°\}$. The values for λ are $\{1.41, 2.82\}$. ϕ should be set to 0 for creating a symmetric filter or $-\frac{\pi}{2}$ for an anti-symmetric filter. These parameters were determined from empirical results.
The filtering produces feature images and the data in these images can be used directly as features for texture discrimination. In addition, each feature image F_k can be processed further for feature extraction. The following features can be derived for a point in F_k centered within a square window of width W

$$\text{ENER}_2 = \frac{1}{N}\sum_{x,y\in W}\sqrt{[F_k^{odd}(x,y)]^2 + [F_k^{even}(x,y)]^2} \quad (14)$$

where N is the number of points in the region W. F_k^{odd} and F_k^{even} refer to anti-symmetric and symmetric filter reponses respectively. The literature suggests using Gaussian weighted windows in the above computation [32, 33]. We use a Guassian with spatial extent 1.3 times larger than the Gabor wavelength.

Markov Random Fields (MRF): A Markov random field texture model represents the global intensity distribution of an image as the joint probability distribution of local conditional systems of each pixel in the image. The image intensity pixel depends only on a set of neighbourhood pixels.
In order to model a texture region using MRFs the Markov process, for a pixel c in image X, is described by a symmetric difference equation [34]:

$$X(c) = \sum \beta_{c,m}[X(c+m) + X(c-m)] + e_c \quad (15)$$

where e_c is a zero mean Gaussian distributed noise (estimation error), m is an offset from the center cell c and $\beta_{c,m}$ are parameters that weigh a pair of symmetric neighbours to the center cell. The βs form the features vector that describes the Markovian properties of the texture and govern the spatial interactions. A region R of size $w \times w$ is defined together with the order of the neighbourhood. The order describes the set of pairs of symmetrical neighbours, and their offsets m from center pixel c, that interact with the center pixel. For every pixel c in R, its neighbouring pixels up to the specified order describe a spatial interaction with the pixel. These spatial interactions for all c in R are modelled using the Gauss model described above. We can represent Eq. (15) in matrix notation:

$$X(c) = \beta^T Q_c + e_c \quad (16)$$

where β^T is a vector consisting of all the $\beta_{c,m}$ and Q_c is a vector defined by:

$$Q_c = \begin{bmatrix} X(c+m_1) + X(c-m_1) \\ X(c+m_2) + X(c-m_2) \\ X(c+m_3) + X(c-m_3) \\ \cdots \end{bmatrix} \quad (17)$$

The βs are estimated using a least squares approach. Features for a region R are computed by a modified method based on the one presented by Cesmeli and Wang [35]:

$$\mathrm{AAD}_2 = \frac{1}{w^2} \sum_{c \in R} \mid X(c) - \beta_j Q_{c_j} \mid \qquad (18)$$

where w^2 is the number of points in R. The neighbourhood order is 2, which has 4 pairs of symmetric neighbours. Hence, the number of texture features is 4. The region R is 9×9 in size.

Parameter estimation and features extraction: The choice of parameters for texture description is important for accurate image segmentation. Parameters for the implemented methods are estimated empirically. The feature images produced from the feature extraction process are filtered (using local averaging of a 5×5 window) to remove sharp spikes. Texture features are normalized so that the sample distribution for each image has zero mean and unit variance. Table 1 summarizes the texture description parameters.

Region merging: Clustering is used to group similar pixels together based on their computed texture features. This creates homogenous regions in an image that can be segmented. Fuzzy clustering allows data to belong to more than one class [36, 37]. This is reflected by their degree of membership in a particular cluster. It is based on the minimization of the objective function

$$J_m = \sum_{j=1}^{C} \sum_{i=1}^{N} u_{ij}^m \parallel \mathbf{x}_i - \mathbf{c}_i \parallel^2 \ , 1 \leq m \leq \infty \qquad (19)$$

where m, the fuzzy factor, is a real number greater than 1, u_{ij} the degree of membership of $\mathbf{x}_i$ in the cluster j and $\mathbf{x}_i$ is the ith d-dimensional data. The d-dimensional center of a cluster is denoted by $\mathbf{c}_j$ and $\parallel * \parallel$ is the norm. The variables C and N refer to the number of clusters and the number of members per cluster respectively.
Fuzzy partitioning is an iterative optimization process. The membership u_{ij}, in matrix U, and the cluster centers $\mathbf{c}_j$ are computed by

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{\|\mathbf{x}_i - \mathbf{c}_j\|})^{\frac{2}{m-1}}} \qquad (20)$$

$$\mathbf{c}_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot \mathbf{x}_i}{\sum_{i=1}^{N} u_{ij}^m} \qquad (21)$$

The algorithm terminates when $\mid u_{ij}^{k+1} - u_{ij}^k \mid$ is less than ϵ, where ϵ is a threshold between 0 and 1 and k an iteration step.

Classifier design: The segmentation algorithm that is implemented uses supervised classification to identify image regions created by the fuzzy clustering algorithm. A training phase incorporates *a priori* knowledge into a classifier. Ideally, the training images will effectively represent the structure of the feature space of unknown samples. A common issue in supervised classifier design is the presence of a large number of pattern samples that make the training process extremely inefficient. We first perform sample selection before computing classifier parameters.

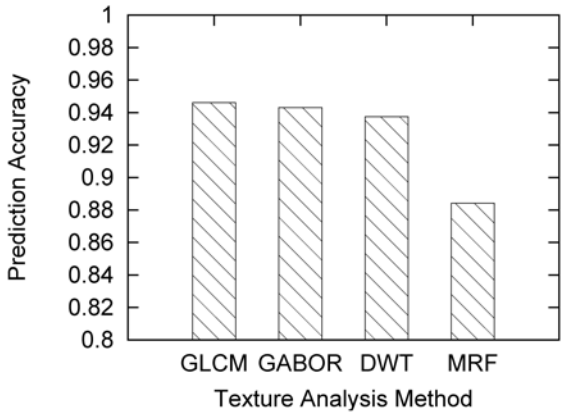


Figure 3: Average prediction accuracy

20 images are selected for training samples. In these images, the different image regions are labelled, denoted as IRIS, PUPIL, REFLECTION, SKIN and EYELASH. The labelled image and the original image form the input to an algorithm that computes texture features and then uses the labelled image to construct sets of feature vectors belonging to the image regions mentioned above. These pattern sets are then reduced separately, recombined and evaluated for information content.

A sample set f is reduced by removing redundant samples using a KNN algorithm [38]. A k value is first specified and then the algorithm proceeds by selecting a pattern $\mathbf{x}$ from f and discarding its k nearest neighbours in f. The selected feature vector is placed in a new set f_{new}. The initial set f will now contain neither $\mathbf{x}$ nor its k nearest neighbours. This process is repeated on f until it is empty. From an empirical evaluation, we use $k = 20$ which selects approximately 2500 vectors (about 5% of the total feature set in the image) - this eases the computational burden and also reduces the risk of overtraining. The information content of the selected samples is measured by considering their ability to predict labels of the initial sample set.

Given an unlabelled image pixel and its feature vector from the initial sample set, we assign to the pixel the label of the vector in the reduced set that is nearest to it (the Euclidean distance is used for feature comparison). The predicted labels are then compared to the actual class labels and a normalized prediction accuracy is computed - 1 denotes the ability to predict all labels correctly; 0 denotes no labels being predicted correctly at all.

Figure 3 shows the average predication accuracy for the 20 training images. The accuracies for the GLCM, GABOR and DWT methods are similar. However, the MRF accuracy is significantly lower. This highlights

Table 1: Summary of texture parameters

Method	Parameters	Features	Window size	#Features
GLCM	$\theta \in \{0°, 45°, 90°, 135°\}$ $d \in \{1\}$	CONT, ENTR $MEAN_1$, SDEV	9×9	16
DWT	2 passes	Original image - $MEAN_2$, AAD_1 Each pass - LL: $MEAN_2$, AAD_1 LH, HL: $ENER_1$	8×8	10
GABOR	$\gamma = 0.5, B_f = 1.0, B_\theta = 30°$ $\theta \in \{0°, 30°, 60°, 90°, 120°, 150°\}$ $\lambda \in \{1.41, 2.82\}$	$ENER_2$	Dependent on λ	12
MRF	Order=2	AAD_2	9×9	4

the fact that more samples are required, as compared to the other methods, to effectively represent its pattern set. Since we are using a KNN approach to remove redundant samples, we can conclude that the feature vectors of the GLCM, GABOR and DWT are generally more compact than the MRF. The KNN algorithm will retain important information if clusters are compact since a selected feature vector has had neighbours discarded that are very close to it in feature space. In the case of the MRF, a pattern not similar to the selected one but included in the set of *k* nearest neighbours will be discarded. Compactness is highly desirable for homogeneity. Our method of subset selection of samples provides a good estimate of the original feature space since 88 to 95% of the patterns can be correctly predicted for the different texture analysis methods. The selected samples are used to estimate parameters for the classifier.

A classifier that uses the 2 class Fisher linear discriminant is implemented for region classification [11]. The selected sample sets for the 20 training images and their corresponding labels are used to compute a discriminant for every combination of two classes (producing 10 discriminants). This is done for each texture method to complete the learning process. Given the five image region classes (IRIS, PUPIL, REFLECTION, SKIN and EYELASH) we have 10 two category combinations e.g. IRIS-EYELASH, REFLECTION-PUPIL, etc. For each pair of categories, the Fisher method computes a weight vector **w** for the separating plane. As such, an *n*-dimensional sample is projected to a single dimension.

The 1-dimensional Fisher projections for each class pair are then modelled as two Gaussians (one for each class). For each class of projected 1-dimensional points, the mean μ and standard deviation σ are computed. The distribution for the set of points is assumed to be the univariate Gaussian function

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-0.5\left(\frac{x-\mu}{\sigma}\right)^2\right] \qquad (22)$$

Hence, for each pair there are 2 probability distributions. Given these 2 distributions, we use Bayesian decision theory to classify a feature. A pattern **x** is classified into class ω_i if

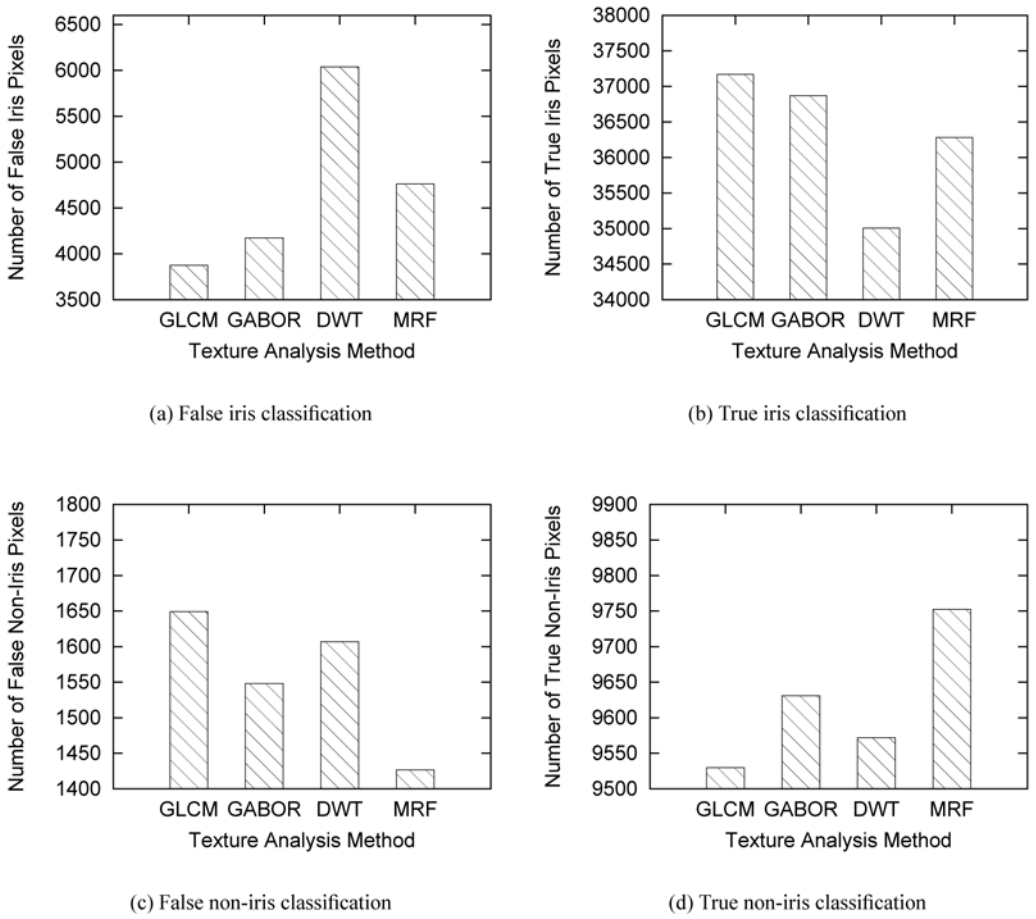
$$P(\omega_i \mid \mathbf{x}) = max(P(\omega_1 \mid \mathbf{x}), P(\omega_2 \mid \mathbf{x})) \qquad (23)$$

where $P(\omega \mid x)$ can be determined using Eq. (22) for a particular class. For the ten discriminants, there are ten outputs. The class label of a feature is defined as being the most common label in the ten outputs. In the case of a tie, a class is assigned randomly from those in the tie set. The final regions are renamed such that PUPIL, REFLECTION, SKIN and EYELASH are denoted NON-IRIS.

Iris segmentation using pattern classification: Once features are extracted from an image, they are either used as input to the training algorithm or the image segmentation algorithm. The method of segmentation that we use is region growing. Regions are grown by clustering the feature vectors, where each cluster represents an image region. Image regions are identified by classifying the cluster centroids using the Fisher linear discriminant. The clustering algorithm has parameters that need to be supplied and a stopping criterion to be defined. For simplicity and completeness, it is assumed that the stopping criterion for the iterative process is a binary function s that uses the change in centroids at the current iteration and the previous iteration to decide whether to proceed or not.
Let the centroids at step k be $C_k = \{\mathbf{c}_k^1, \mathbf{c}_k^2, \ldots, \mathbf{c}_k^n\}$ and let the centroids at step $k-1$ be $C_{k-1} = \{\mathbf{c}_{k-1}^1, \mathbf{c}_{k-1}^2, \ldots, \mathbf{c}_{k-1}^n\}$ where n is the number of centroids. Then

$$s(C_K, C_{K-1}) = \begin{cases} 0, \mid \mathbf{c}_k^i - \mathbf{c}_{k-1}^i \mid \leq \epsilon, \ \forall i = 1 \ldots n \\ 1, \text{otherwise} \end{cases} \qquad (24)$$

The clustering process runs as long as s evaluates to 1. A maximum number of iterations (80) is also introduced so that the process is not exhaustive. We use function s for the fuzzy clustering since evaluating a stopping criterion for the membership matrix U is computationally burdensome. The fuzzy factor m is 2 and $\epsilon = 0.0001$. The number of classes for the clustering algorithm is 6. More details regarding computing the number of clusters can be found in [11]. The Euclidean distance is used as a measure of pattern similarity.



(a) False iris classification

(b) True iris classification

(c) False non-iris classification

(d) True non-iris classification

Figure 4: Test results

3. EXPERIMENTAL RESULTS AND DISCUSSION

100 test images were selected from the CASIA image database for experimental analysis. Ground truths for these input images were created manually by a human operator. The segmentation algorithm described in the previous section is executed 4 times, each time using one of the 4 texture descriptors, for each input image. Since the fuzzy clustering algorithm is randomly initialized, there is a possibility that it settles in a local optimum. Therefore, we perform 10 clustering test runs for each texture descriptor and then compute the average number of true and false pixel classifications for the 2 classes (IRIS and NON-IRIS) for each image; the number of pixels in an input iris image is 52224. The 100 segmentation results for each method are then averaged to produce the final outcome.

Figure 4 shows the results obtained. In order to evaluate the performance of each method, we considered the segmentation accuracy at the finest level i.e. pixels. This way, distinctions could easily be made since the segmentation accuracy viewed as a percentage for each class type does not appear significant across the different methods. We summarize the results:

- **IRIS:** GLCM performs the best for segmenting IRIS texture. On average, it segments 300 pixels more than the next best method (GABOR). The GABOR performs second best for this texture class. The MRF, segmenting 589 pixels less than GABOR, lies in third place. DWT performs the worst amongst the four methods for segmenting IRIS texture. On average, it segments 2163 pixels fewer than the GLCM.
- **NON-IRIS:** MRF provides the best discrimination for NON-IRIS pixels. However, as seen above, it performs poorly on IRIS pixels. The GABOR functions perform second best for this class of pixels (segments 121 pixels less than MRF). The DWT and GLCM rank third and fourth respectively for this class of pixels. The GLCM, while providing the best segmentation for IRIS pixels, provides the worst performance for the NON-IRIS class.

Table 2 provides a summary of Figure 4 by listing the segmentation accuracy as a percentage. From a global point of view, the four methods differ by 1-5% in their abilities to segment each class. While this doesn't appear significant, one must keep in mind that the NON-IRIS artifacts tend to be small in size compared to the rest of

Texture Method	False Non-iris (%)	False Iris (%)	True Non-iris (%)	True Iris (%)
GLCM	14.8	9.4	85.2	90.6
GABOR	13.8	10.2	86.1	89.8
DWT	14.4	14.7	85.6	85.3
MRF	12.8	11.6	87.2	88.4

Table 2: Summary of results

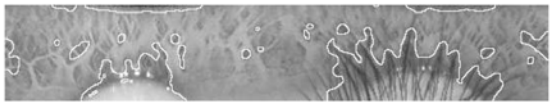
the image. The observations above put this into perspective by showing that these measurements fall in the region of a few to several hundred pixels. This is sufficient to rank the different methods for quality of segmentation. Another observation is that no single method provides the best discrimination for IRIS and NON-IRIS classes. This highlights the fact that a texture feature set is suitable for a particular *type* of texture and cannot be expected to perform equally well with another type.

In Figure 5(a) and Figure 5(b), a normalized iris image and its segmentation, using the GLCM texture descriptor, is presented. Small regions, falsely classified as NON-IRIS texture, can be seen in the result. A connected components algorithm can be used to improve the segmentation by removing these components in the image [7]. This is shown in Figure 5(c).

Several false classifications for a GABOR filter output are shown in Figure 6. This also occurs in some cases when applying the other techniques. Several explanations for this occurrence are possible: i) the window size for computing texture features is too small, ii) the texture features themselves are poor descriptors and iii) the Fisher linear discriminant cannot easily separate the 2 types of classes.

The segmentation algorithm classifies and labels the image pixels into one of two categories - iris or non-iris. Using these labels, the iris texture is extracted, as shown in Figure 7. This is the final segmentation of an iris image. Most of the artifacts in the image are removed. Although the segmentation is not 100% accurate, it is very effective. The eyelashes, eyelids, reflection and pupil pixels have been removed to a great extent. The feasibility of a feature extraction and pattern classification approach for segmenting iris images has been demonstrated.

(a) Input iris image

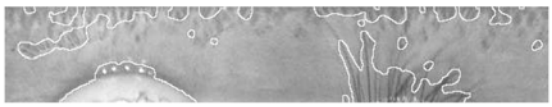


(b) Segmented image

(c) Connected components filtering

Figure 5: Segmentation result using GLCM texture features

(a) Input iris image

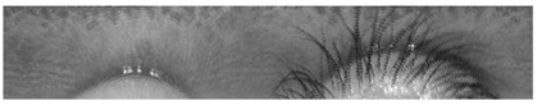


(b) Segmented image

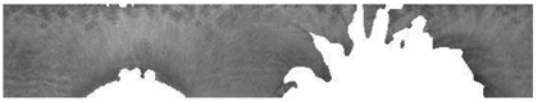
Figure 6: Segmentation using GABOR features

4. CONCLUSIONS

In this paper, we focussed on the classic problem of image segmentation. Texture analysis and pattern recognition techniques are used to extract useful iris texture from an image of the eye. To our knowledge, this method of iris segmentation is new to the field and we have developed a feasible solution that provides good performance. The experimental results show that no single method tested in this paper can provide the best discrimination for the IRIS and NON-IRIS classes. The GABOR kernels provide good segmentation for both the IRIS and NON-IRIS classes; the GLCM method has the highest number of true classifications for IRIS pixels while the MRF performs



(a) Input iris image



(b) Extracted iris texture

Figure 7: Extracted iris texture

similarly for NON-IRIS pixels. The DWT provides fair performance but it has the lowest accuracy amongst all the methods.

The feature sets used used in this study may not be ideal for iris texture description. Several other methods are present in the literature for further study e.g. Fourier descriptors, fractals and other wavelet functions. Optimal feature sets can be determined by using combinations of different features and assessing their segmentation or classification accuracy. Alternate techniques for constructing homogenous image regions, such as split-and-merge and the watershed algorithm, can also be found in the literature [7]. Many different pattern classification techniques are also available for study [38].

REFERENCES

[1] F. Adler, *Physiology of the eye*. St. Louis: MO:Mosby, 1965.

[2] W. Kong and D. Zhang, "Detecting eyelash and reflection for accurate iris segmentation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 6, pp. 1025–1034, 2003.

[3] J. Huang, Y. Wang, T. Tan, and J. Ciu, "A new iris segmentation method for recognition," in *International Conference on Image Processing*, vol. 3, 2004, pp. 554–557.

[4] L. Ma, T. Tan, Y. Wang and D. Zhang, "Personal identification based on iris texture analysis," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 25, no. 12, pp. 1519–1533, 2003.

[5] "Casia Iris Database version 1." [Online]. Available: http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp

[6] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov 1986.

[7] R. Gonzalez and R. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, 2002.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.

[9] J. Daugman, "High confidence visual recognition of persons by a test of statistical independance," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 15, no. 11, pp. 1148–1961, 1993.

[10] E.H. Land, "Recent advances in retinex theory," *Vision Research*, vol. 26, no. 1, pp. 7–21, 1986.

[11] A. Bachoo, "Comparison of segmentation methods for an accurate iris extraction," Master of Science, School of Computer Science - University of KwaZulu-Natal, December 2006.

[12] B. Jähne, H. Haußecker and P. Geißler, *Handbook of Computer Vision and Applications - Volume 2*. Academic Press, 1999.

[13] M. Sonka, V. Hlavac and R. Boyle, *Image Processing: Analysis and Machine Vison*. PWS Publishing Company, 1999.

[14] A. Rosenfeld and A. Kak, *Digital picture processing*. Academic Press, 1982, vol. 1.

[15] A.K. Jain, *Fundamentals of digital image processing*. New Jersey: Prentice-Hall, 1989.

[16] C.H. Chen, L.F. Pau and P.S.P. Wang, *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*. World Scientific Publishing Co., 1998.

[17] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[18] R.M. Haralick, K. Shanmugam and I. Dinstein, "Texture features for image classification," *IEEE Trans. System Man. Cybernat.*, vol. 8, no. 6, pp. 610–621, 1973.

[19] J.S. Weszka and A. Rosenfeld, "An application of texture analysis to material inspection," *Pattern Recognition*, vol. 8, pp. 195–199, 1976.

[20] L.S. Davies, S.A. Johns and J.K. Aggarwal, "Texture analysis using generalized co-occurence matrices," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 1, no. 3, pp. 251–259, 1979.

[21] M.M. Trivedi, R.M. Haralick, R.W. Conners and S. Goh, "Object detection based on grey level co-occurrence," *Computer Vision, Graphics and Image Processing*, vol. 28, pp. 199–219, 1984.

[22] R.W. Conners and C.A. Harlow, "A theoretical comparison of texture algorithms," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. PAMI-2, pp. 204–222, 1980.

[23] Q. Zhang, J. Wang, P. Gong and P. Shi, "Study of urban spatial patterns from spot panchromatic imagery using textural analysis," *Int. J. Remote Sensing*, vol. 24, no. 21, pp. 4137–4160, 2003.

[24] S. Mallat, "A theory of multiresolution signal decomposition," *IEEE Trans. on Pattern Analysis and Machine Inteligence*, vol. 11, no. 7, pp. 674–693, 1989.

[25] M. Kokare and B. Biswas, "Texture image retrieval using rotated wavelet filters," *Pattern Recognition Letters*, no. 28, pp. 1240–1249, 2007.

[26] C.-W. Liang and p.-Y. Chen, "DWT based text localization," *International Journal of Applied Science and Engineering*, vol. 2, no. 1, pp. 105–116, 2004.

[27] J.-L. Starck, F. Murtagh and A. Bijaoui, *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press, 1998.

[28] D. Clausi, "Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery," *Atmosphere-Ocean*, vol. 39, no. 3, pp. 183–194, 2001.

[29] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[30] S. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Trans. on Image Processing*, vol. 11, no. 10, pp. 1160–1167, 2002.

[31] P. Kruizinga and N. Petkov, "Nonlinear operator for orientated texture," *IEEE*, vol. 8, no. 10, pp. 1395–1407, 1999.

[32] A.K. Jain, N.K. Ratha and S. Lakshmanan, "Object detection using Gabor filters," *Pattern Recognition*, vol. 30, no. 2, pp. 295–309, 1997.

[33] D.A. Clausi and M.E. Jernigan, "Designing Gabor filters for optimal texture separability," *Pattern Recognition*, vol. 33, pp. 1835–1849, 2000.

[34] J. Woods, "Two-dimensional discrete Markovian fields," *IEEE Trans. Info. Theory*, vol. 18, no. 2, pp. 232–240, 1972.

[35] E. Cesmeli and D. Wang, "Texture segmentation using Gaussian-Markov random fields and neural oscillator networks," *IEEE Trans. on Neural Networks*, vol. 12, no. 2, pp. 394–404, 2001.

[36] J. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well separated clusters," *Journal of Cybernetics*, vol. 3, pp. 32–57, 1974.

[37] J. Bezdek, *Pattern Recognition with Fuzzy Object Function*. Plenum Press, 1981.

[38] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern classification*. John Wiley and Sons, 2001.