Iris Recognition – Quality Assessment

Goutam Das

International Institute of Information Technology, Bhubaneswar

Abstract -

[Section 1]

Iris recognition is one of the most reliable biometric systems available today for recognition purposes.

The reason why iris is so reliable because: unlike fingerprint and face, the iris is well protected against damage from external environment, the fine textures in the iris remain stable over many decades. Moreover the commercially deployed iris recognition algorithm has an unprecedented false match rate.

But the performance of the iris recognition can be greatly affected by poor image quality. The proposed method involves calculation of a quality metric to create quality scores of identical iris images in the database such that the image with the highest score has the best matching rate.

Factors such as occlusion, blur and pupil dilation were selected as the ones which mostly attenuate iris images and can result in errors during recognition. The method proposes to create a measure of each factor and according to its importance create a fusion score to define the quality of the image relative to other images.

The observations were tested on the IIT-Database, which comprises of gray-scale iris images of 225 individuals, with 10 images per person(5 left and 5 right).

Introduction and literature survey -

[Section 2]

Almost all iris recognition algorithms which have been proposed till date follow the same structure of recognition: (a) Image acquisition (b)Segmentation (c)Normalization (d)Feature extraction (e)Matching

Off all these steps, segmentation is the most important as it involves removal of the iris from the rest of the eye, from which features can be extracted at a later stage and be used for matching. Thus most of the works published focus on various methods through which proper segmentation can take place.

But the quality of the iris image depends on the image acquired. Poor quality images in most cases, result in improper segmentation or feature extraction.

From the works of [1],the major factors which result in poor quality iris images are- Blur(defocus and motion), Off angle, occlusion, pupil dilation and improper lighting/reflection. The factors like off-angle, blur affect the segmentation while others like occlusion, pupil dilation affect feature extraction.

The proposed method aims to create a quality measure by estimating the following factors-

1-Blur

2-Occlusion

3-Pupil Dilation

4-Reflection

Daugman[2] developed and patented the first iris detection algorithms in 1991. Most of the commercial iris biometric systems used today use his algorithms. After that several other methods have been proposed by Wildes[3], Li Ma[4] and others, providing new methods of recognition.

Quality assessment has been performed by some authors in the past. ND Kalka et al.[1] studied the effects of defocus blur, motion blur and off-angle on iris images and fused the individual quality scores of these factors using the Dempster-Shafer criterion to produce a single score.

Before them, many previous works had been published like the ones from Zhu[5],Chen et al.[6], Wei et al.[7] and Daugman[8], but these works mostly focused on the estimation of a single or a pair of factors.

Craig Belcher[9] also proposed a selective feature approach for image quality measure. This work consisted of an estimation of occlusion and pupil dilation and calculate a fusion score

The rest of the report is organized as follows: Section 3(a) contains the algorithms used for iris recognition while Section 3(b) contains a study of estimation of various quality factors. Section 4 lists observations of the ideas discussed in 3(b) and Section 5 concludes with the results and expected future work.

Proposed Work

[Section 3(a)]

Before creating a quality metric for quality assessment of iris images, an iris recognition algorithm needs to be implemented to test the results for different quality of images.

In the proposed method, Daugman's algorithms[2] were implemented for segmentation and matching purposes. Each step of the recognition process is as follows:

Segmentation -

Segmentation or localization is the removal of the iris region from the rest of the image. The acquired image of the eye usually contains the iris, sclera, pupil, eyelids and eyelashes. The contours of the iris were initially assumed to be circular in nature, but later works considered an elliptical boundary.

For the current project, Daugman's integro-differential operator was used to identify the boundaries of both the pupil and the iris boundaries. The integro-differential operator can be defined as

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$$

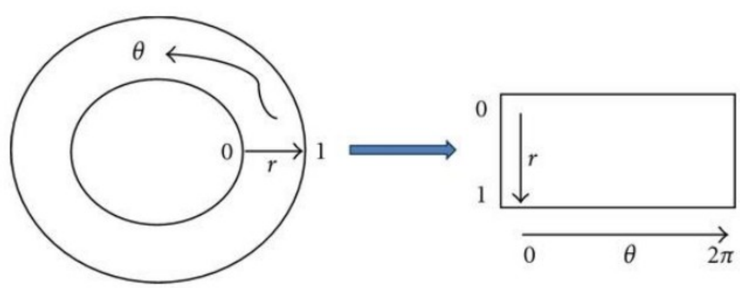
where I(x,y) is the iris image. The operator searches over the image domain(x,y) for the maximum in the blurred partial derivative with respect to increasing radius r, of the normalized contour of integral of I(x,y) along the circular arcs of 'ds' of radius r and center coordinates (x0,y0).

After successful localization of the iris, the eyelids and eyelashes can be removed by using either linear hough transform or the modified integro-differential operator to detect the eyelids some per-processing to remove the eyelashes.

Normalization -

Normalization was performed using Daugman's Rubber Sheet model. In this method, the Cartesian coordinates of the segmented iris are converted to polar coordinates as follows

Transformation of cartesian to polar coordinates



Formula -

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta)$$

$$x(r,\theta) = (1-r)x_p(\theta) + rx_l(\theta)$$

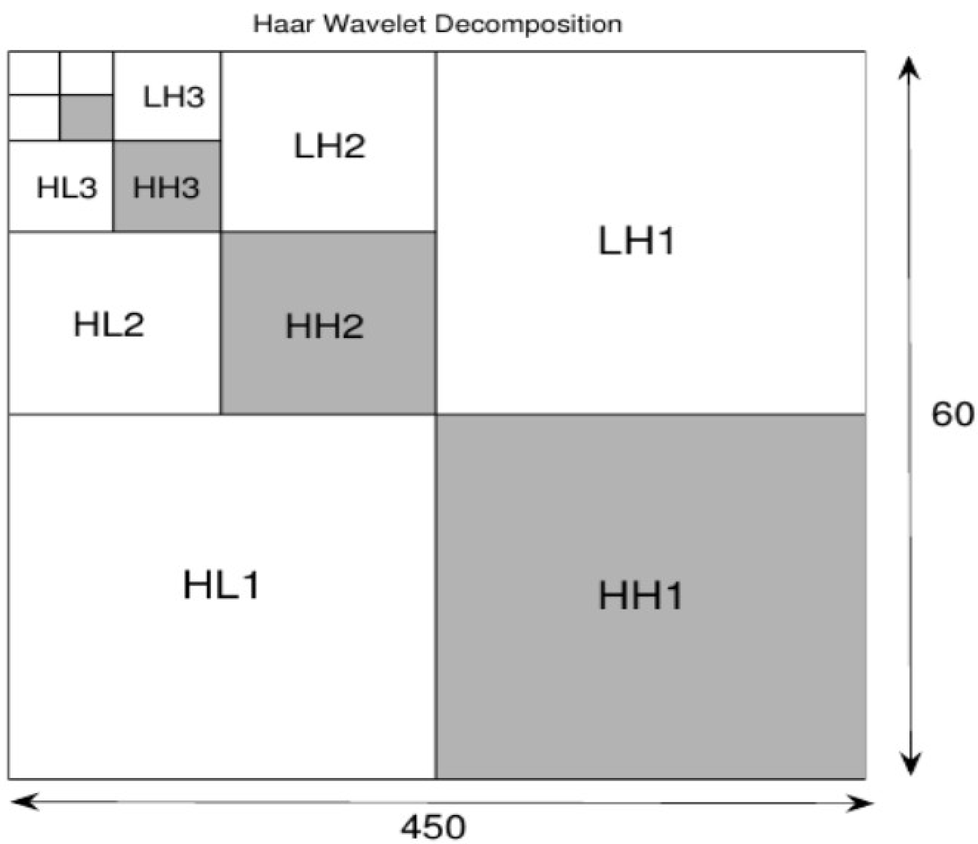
$$y(r,\theta) = (1-r)y_p(\theta) + ry_l(\theta)$$

The value of theta ranges from 0 to 360 degrees, while 'r' is from 0 to 1

Feature Extraction -

After successful normalization of the iris, feature extraction needs to be performed to extract the unique features of the iris for storage or matching purposes.

5 level haar wavelet transform was applied on the normalized iris image. From this, the vertical, horizontal and diagonal features from the first the 4^{th} and 5^{th} levels of the transform were used as the required features for matching purposes.



Matching -

The extracted features were then converted to iris codes as follows:

$$\text{iris code} = \begin{cases} 0 & \text{if coeff} < 0 \\ 1 & \text{if coeff} > 0 \end{cases}$$

The iris codes generated were then matched using the Hamming Distance which can be defined as follows -

$$HD = \frac{1}{N} \sum_{i=1}^{N} X_i \oplus y_i$$

Where X_i and Y_i represent the i^{th} bit in the iris code sequence and X and Y, and N is the total number of bits in each iris code sequence. The symbol $\oplus$ is the "XOR" operator.

QUALITY MEASURE -

[Section 3(b)]

To create a metric for quality assessment, each factor needs to be estimated individually.

For this work, precise segmentation of the iris image is not required and only a rough segmentation method is implemented to localize the iris and assess its quality.

Thus for factors such as blur and pupil dilation, only the integro differential operator was applied to get a localized iris image. The analysis is performed on the normalized iris region.

Estimation of Blur -

Kalka et al.[1] calculated an estimation of both defocus and motion blur, the former using Daugman's 8x8 convolution kernel[4].

In this work, Min Goo Choi et al.[9] method was used as an estimation of blur. This method creates a measure of blur by calculating the number of blurred edges in the image.

This method was used for estimation of blur as, though highly blurred iris can affect segmentation process, blur mainly affects effective feature extraction as the in blurring, the high frequency components are attenuated which can affect feature extraction.

The algorithm is explained as follows:

(1) The normalized iris image is taken as the input image. The test image is denoted with M rows and N columns as

$$f(x,y), \text{ for } x \in [1,M] \text{ and } y \in [1,N].$$

(2)The Horizontal absolute difference value of each pixel in the image is calculated, where the HADV is given as -

$$D_h(x,y) = \left| f(x,y+1) - f(x,y-1) \right|.$$

(3) The mean(Dh_{mean}) of HADV for the whole image is calculated -

$$D_{h-mean} = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} D_h(x,y).$$

(4)If the HADV value is larger than $Dh_{mean,}$ the pixel becomes an edge candidate $ch_{(x,y)}$

$$C_h(x,y) = \begin{cases} D_h(x,y) & \text{if } D_h(x,y) > D_{h-mean} \\ 0 & \text{otherwise} \end{cases},$$

(5)Detecting Horizontal edge pixels

$$E_h(x,y) = \begin{cases} 1 & \text{if } C_h(x,y) > C_h(x,y+1) \text{ and } \\ & C_h(x,y) > C_h(x,y-1) \\ 0 & \text{otherwise} \end{cases}$$

(6) Now to determine whether the edge pixel is blurred or not. The ratio value of for blur decision is obtained horizontally

$$A_h(x,y) = \frac{1}{2} D_h(x,y),$$

$$BR_h(x,y) = \frac{|f(x,y) - A_h(x,y)|}{A_h(x,y)}.$$

(7) In the same way, the estimate BR_v in the vertical direction can be calculated by repeating the previous steps. The larger value between Brh and Brv is selected for the final decision, which is called inverse bluriness.

$$B(x,y) = \begin{cases} 1 & \text{if } \max(BR_h(x,y), BR_v(x,y)) < Th_B \\ 0 & \text{otherwise} \end{cases}.$$

By experiment, blur is best detected when Thb is 0.1

(8) Finally, the mean of blur and ratio of the edge is calculated.

$$Blur_{ratio} = \frac{Blur_{cnt}}{Edge_{cnt}}$$

Estimation of Pupil Dilation -

Pupil dilation was measured by taking the ratio of the pupil and iris radius.

As this work assumes the iris and pupil to be circles and uses the integro-differential operator to detect the center and radius of these circles(outer boundaries of the pupil and the iris), the same values were used to estimate the pupil dilation(D) as follows -

$$D = \frac{\text{Pupil radius}}{\text{Iris Radius}} \times 100\%.$$

Estimation of Occlusion -

Occlusion refers to the blocking of the iris by the eyelids and eyelashes.

Most of the acquired iris images contain some level of occlusion, but it usually does not affect the recognition process. Many algorithms have been developed which can effectively detect the regions of eyelids and eyelashes. During matching, these regions masked and ignored.

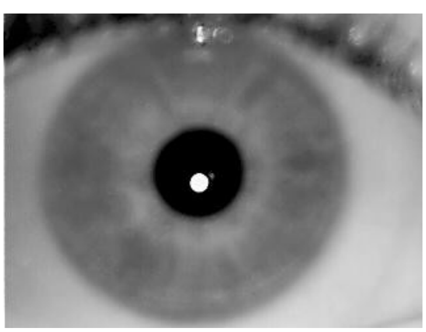
But if the level of occlusion is high, though segmentation can be applied, many important features are lost during extraction and thus effective matching does not occur.

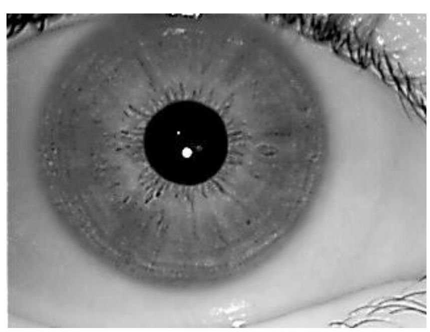
Thus this work proposes to calculate a measure of occlusion only if the level of occlusion is more than 40% of a non-occluded iris image with same dimensions of the pupil and iris.

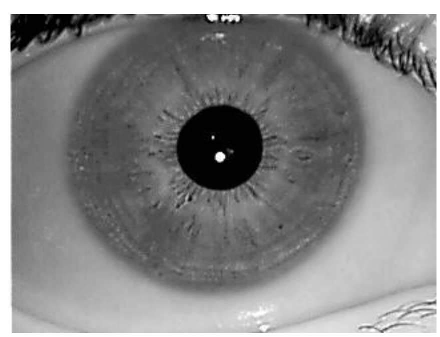
Observations -

The estimation of blur was tested on each individual in the database(225 persons each having 5 left and 5 right iris images). From the results, it was found that it provides a good score to detect the relative level of blur between identical iris images i.e. belonging to the same individual. Example-

IITD Database/003/







Score:0.46683 Score:0.34760 Score:0.38107

Conclusion and Future Works -

[Section 5]

From the result of the tests, it was found that the implemented algorithms provide a good estimation of the level of blur and pupil dilation relative to identical images of the database.

For the estimation of occlusion, though tested methods methods from previous works such as the modified integro-differential operator and others can be used for detection of the areas occluded by eyelids and eyelashes, most of these methods are computationally expensive, and further such level of precision though appreciable is not required to calculate the level of occlusion.

Thus the final quality score is yet to be calculated and can be done so only after an effective method is devised to estimate the occlusion level.

To sum up the future works to be performed are as follows -

(1)An effective segmentation process to calculate the level of occlusion and subsequently, an occlusion score, if the level is high.

(2)Fusion of all estimated scores to a single quality score, and assessment of image quality based on this score.

References

[Section 6]

[1] N.D. Kalka, J.Zuo, N.A.Schmid and B.Cukic, "Image quality assessment for iris biometric" Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on (Volume:40 , Issue: 3)

[2] Daugman, J. "How iris recognition works" Image Processing. 2002. Proceedings. 2002 International Conference on Year: 2002, Volume: 1

[3] Wildes, R.P. "Iris recognition: an emerging biometric technology" Proceedings of the IEEE Year: 1997, Volume: 85, Issue: 9

[4] Yan Li; Wen Li; Yide Ma, "Accurate Iris Location Based on Region of Interest" Biomedical Engineering and Biotechnology (iCBEB), 2012 International Conference on Year: 2012

[5] X. Zhu, Y. Liu, X. Ming, and Q. Cui, “A quality evaluation method of iris images sequence based on wavelet coefficients in ‘region of interest’,” in Proc. 4th Int. Conf. Comput. Inf. Technol., Sep. 2004, pp. 24–27

[6] Y. Chen, S. Dass, and A. Jain, “Localized iris quality using 2-D wavelets,” in Proc. Int. Conf. Biometrics, Hong Kong, 2006, pp. 373–381

[7] Z. Wei, T. Tan, Z. Sun, and J. Cui, “Robust and fast assessment of iris image quality,” in Proc. Int. Conf. Biometrics
, 2006, pp. 464–471.

[8] J. Daugman, “New methods in iris recognition,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 37, no. 5, pp. 1167–1175, Oct. 2007

[9] Min Goo Choi, Jung Hoon Jung, and Jae Wook Jeon “No-Reference Image Quality Assessment using Blur and Noise”, World Academy of Science, Engineering and Technology, Vol:3 2009-02-20

MATLAB CODE -

(1) Iris Recognition (Individual functions for segmentation, normalization are below)

```
%read the image (may be in rgb or grayscale)

img=imread('C:\Users\hcilab\Downloads\10.bmp');

%convert to graysacle and then to double

im1=rgb2gray(img);

im1=im2double(im1);


%Pre-processing

%remove regions of light reflections


%1-complement the image

im2=imcomplement(im1);
```

```
%2-fill in the holes\
im2=imfill(im2,'holes');
%3-complement back to original
im2=imcomplement(im2);

%enhance contrast of image using histogram equalization
im3=histeq(im2);

%apply Daugman's integrodifferential operator
%use the thresh function
rimin=80;%minimum radius of the iris
rimax=120;%maximum radius of the iris
[ci,cp,o]=thresh(im3,rimin,rimax);
imshow(o);

%we get two concentric circles i.e the iris-pupil and iris-sclera boundary
%Extraction of the required area from the image
%iris center and radius
xi=ci(1);
yi=ci(2);
ri=ci(3);
%pupil center and radius
xp=cp(1);
yp=cp(2);
rp=cp(3);

%check if the coordinates of the image are in the concentric circles formed
%or not. Only consider those coordinates which are inside this area.
[si,sj]=size(im1);
```

```
I=zeros(si,sj);
for i=1:si
    for j=1:sj
        if (i-xi).^2 + (j-yi).^2 < ri.^2 && (i-xp).^2 + (j-yp).^2 > rp.^2
            I(i,j)=im1(i,j);
        end
    end
end
%final extracted iris
figure;imshow(I,[]);
I1=I((xi-ri):(xi+ri),(yi-ri):(yi+ri));

%testing
imR=I1;
rMin=0.4;
rMax=1;
M=64;
N=256;
imP = ImToPolar(imR ,rMin, rMax ,M, N);
figure;
imshow(imP,[]);

[r c]=size(imP);
title('Normalized');
E=imP;

for i=1:r
    count=0;
    for j=1:c
```

```
        if E(i,j)~=0
        count=count+1;
        end
        row(i)=count;
    end
end



flag=0;
for i=1:r
    if flag==0 & row(i)>=40

        x1=i;
        flag=1;
    end
     end

%figure;
E1=(E(x1:r,1:c));
%imshow(E1,[]);
% fname=strcat('D:\test\Result\',num2str(inew),'\',num2str(jnew),'.mat');
% save(fname,'E1');
[rE cE]=size(E1);
CW=zeros(size(E1));
for i= 1:cE
    for j=1:rE
      CW(1:j,i)=dct(E1(1:j,i));
      M(1,j)=max(CW(1:j,i));
    end
```

```
    M1(1,i)=max(M);
end
a=1:1:200;
% figure;

%Q=plot(a,M1,':r.');
%hold on;



%testing 2d haar wavelet
[a,h,v,d]=dwt2(E,'haar');
[a1,h1,v1,d1]=dwt2(a,'haar');
[a2,h2,v2,d2]=dwt2(a1,'haar');
[a3,h3,v3,d3]=dwt2(a2,'haar');
[a41,h4,v4,d4]=dwt2(a3,'haar');
a41;
```

1(a) Segmentation using Daugman's Integro-differential operator

```
%function to search for the centre coordinates of the pupil and the iris
%along with their radii
%It makes use of Camus&Wildes' method to select the possible centre coordinates first
%The method consist of thresholding followed by
%checking if the selected points(by thresholding)
%correspond to a local minimum in their immediate(3*s) neighbourhood
%these points serve as the possible centre coordinates for the iris.
%Once the iris has been detected(using Daugman's method);the pupil's centre coordinates
%are found by searching a 10*10 neighbourhood around the iris centre and varying the radius
```

```
%until a maximum is found(using  Daugman's integrodifferential operator)
%INPUTS:
%I:image to be segmented
%rmin ,rmax:the minimum and maximum values of the iris radius
%OUTPUTS:
%cp:the parametrs[xc,yc,r] of the pupilary boundary
%ci:the parametrs[xc,yc,r] of the limbic boundary
%out:the segmented image



function [ci,cp,out]=thresh(I,rmin,rmax);
scale=1;
%Libor Masek's idea that reduces complexity
%significantly by scaling down all images to a constant image size
%to speed up the whole process
rmin=rmin*scale;
rmax=rmax*scale;
%scales all the parameters to the required scale
I=im2double(I);
%arithmetic operations are not defined on uint8
%hence the image is converted to double
pimage=I;
%stores the image for display
I=imresize(I,scale);
I=imcomplement(imfill(imcomplement(I),'holes'));
%this process removes specular reflections by using the morphological operation 'imfill'
%I=nbdavg(I);
%blurs the sharp image formed as a result of using imfill
```

```
rows=size(I,1);
cols=size(I,2);
[X,Y]=find(I<0.5);
%Generates a column vector of the image elements
%that have been selected by tresholding;one for x coordinate and one for y
s=size(X,1);
for k=1:s %
    if (X(k)>rmin)&(Y(k)>rmin)&(X(k)<=(rows-rmin))&(Y(k)<(cols-rmin))
            A=I((X(k)-1):(X(k)+1),(Y(k)-1):(Y(k)+1));
            M=min(min(A));
            %this process scans the neighbourhood of the selected pixel
            %to check if it is a local minimum
           if I(X(k),Y(k))~=M
              X(k)=NaN;
              Y(k)=NaN;
           end
    end
end
v=find(isnan(X));
X(v)=[];
Y(v)=[];
%deletes all pixels that are NOT local minima(that have been set to NaN)
index=find((X<=rmin)|(Y<=rmin)|(X>(rows-rmin))|(Y>(cols-rmin)));
X(index)=[];
Y(index)=[];
%This process deletes all pixels that are so close to the border
%that they could not possibly be the centre coordinates.
N=size(X,1);
%recompute the size after deleting unnecessary elements
```

```
maxb=zeros(rows,cols);

maxrad=zeros(rows,cols);

%defines two arrays maxb and maxrad to store the maximum value of blur

%for each of the selected centre points and the corresponding radius

for j=1:N

    [b,r,blur]=partiald(I,[X(j),Y(j)],rmin,rmax,'inf',600,'iris');%coarse search

    maxb(X(j),Y(j))=b;

    maxrad(X(j),Y(j))=r;

endfor

[x,y]=find(maxb==max(max(maxb)));

ci=search(I,rmin,rmax,x,y,'iris');%fine search

%finds the maximum value of blur by scanning all the centre coordinates

ci=ci/scale;

%the function search searches for the centre of the pupil and its radius

%by scanning a 10*10 window around the iris centre for establishing

%the pupil's centre and hence its radius

cp=search(I,round(0.1*r),round(0.8*r),ci(1)*scale,ci(2)*scale,'pupil');%Ref:Daugman's paper that sets biological limits on the relative sizes of the iris and pupil

cp=cp/scale;

%displaying the segmented image

out=drawcircle(pimage,[ci(1),ci(2)],ci(3),600);

out=drawcircle(out,[cp(1),cp(2)],cp(3),600);
```

1(b) Normalization using Rubber Sheet model

```
function imP = ImToPolar (imR, rMin, rMax, M, N)
```

```
% IMTOPOLAR converts rectangular image to polar form. The output image is
% an MxN image with M points along the r axis and N points along the theta
% axis. The origin of the image is assumed to be at the center of the given
% image. The image is assumed to be grayscale.
% Bilinear interpolation is used to interpolate between points not exactly
% in the image.
%
% rMin and rMax should be between 0 and 1 and rMin < rMax. r = 0 is the
% center of the image and r = 1 is half the width or height of the image.
%



[Mr Nr] = size(imR); % size of rectangular image
Om = (Mr+1)/2; % co-ordinates of the center of the image
On = (Nr+1)/2;
sx = (Mr-1)/2; % scale factors
sy = (Nr-1)/2;

imP = zeros(M, N);

delR = (rMax - rMin)/(M-1);
delT = 2*pi/N;

% loop in radius and
for ri = 1:M
for ti = 1:N
   r = rMin + (ri - 1)*delR;
   t = (ti - 1)*delT;
   x = r*cos(t);
```

```
    y = r*sin(t);
    xR = x*sx + Om;
    yR = y*sy + On;
    imP (ri, ti) = interpolate (imR, xR, yR);
end
end
```

2. Estimation of blur

```
%Estimation of blur using Min Goo Choi et al. method
img=imread('/home/goutam/iris/IITD_Database/003/01.bmp'); %read the image
img=im2double(img);
%f1=fspecial('gaussian',11,5);
%img=conv2(img,f1);
%calculation of the Horizontal Absolute Difference Value(HADV)
[r c]=size(img);
dh=zeros(r,c);
for i=1:r
  for j=2:c-1
    dh(i,j)=abs(img(i,j+1)-img(i,j-1));
  end
end
%calculation of the horizontal mean
dh_mean=mean(mean(dh));
```

```
%calculation for ch(x,y) i.e.the candidates for the edge pixels
%ch(x,y) will be equal to to dh(x,y) if dh(x,y) is greater than dh_mean
ch=zeros(r,c);
for i=1:r
  for j=2:c-1
    if dh(i,j) > dh_mean
      ch(i,j)=dh(i,j);
    end
  end
end
%detecting the horizontally edge pixels
%eh(x,y) is if it is greater than the horizontally adjacent pixels
eh=zeros(r,c);
for i=1:r
  for j=2:c-1
    if ch(i,j) > ch(i,j-1) && ch(i,j) > ch(i,j+1)
      eh(i,j)=1;
    end
  end
end
%calculation of blur, horizontal and vertical
ah=zeros(r,c);brh=zeros(r,c);
av=zeros(r,c);brv=zeros(r,c);
for i=2:r-1
  for j=2:c-1
    ah(i,j)=0.5*abs((img(i,j+1)-img(i,j-1)));
    av(i,j)=0.5*abs((img(i-1,j)-img(i+1,j)));
    if ah(i,j)==0
      brh(i,j)=abs(img(i,j)-ah(i,j));
```

```
    else
      brh(i,j)=abs(img(i,j)-ah(i,j))/ah(i,j);
    end
    if av(i,j)==0
      brv(i,j)=abs(img(i,j)-av(i,j));
    else
      brv(i,j)=abs(img(i,j)-av(i,j))/av(i,j);
    end
  end
end
B=zeros(r,c);
%threshold value
T=0.1;
for i=1:r
  for j=1:c
    if max(brh(i,j),brv(i,j)) < T
      B(i,j)=1;
    end
  end
end
ep=sum(sum(eh));bp=sum(sum(B));
ratio=bp/ep;disp(ratio);
```