

Fine Circular Pupil Localization

Ihab A. Mohammed

Department of Computer Science, College of Science, University of Al-Nahrain, Baghdad-Iraq.

E-mail: eihabmurjan@yahoo.com.

Abstract

In this paper a simple, fast, and accurate method is proposed to localize the pupil. After analyzing eye images and their histograms, it has been found that the pupil area takes about 10% of the eye image area and mainly it is the darkest part. A threshold value was computed from the histogram of the image, the computed threshold is used to convert the eye image to binary image, the seed fill algorithm was used as region growing method to segment the binary image and locate the pupil as the maximum segment, then fill it with black color to remove any specular spot reflection. A circle fitting algorithm is used to locate the best pupil circle. The conducted test on the proposed system using 10 eye images, downloaded from CASIA-IRISV3-Interval database, indicated that the pupil area average error ratio is 0.6%, the pupil center average position error is 0.5%, and the pupil radius average error ratio is 2.1%.

Keywords: Pupil Localization, Histogram, Threshold, Segmentation, Seed Filling, Circle Fitting.

1. Introduction

The eye image consists of a dark circular area known as the pupil with a specular spot reflection inside, a colored texture represents the iris, a white area called the sclera, upper and lower lids, and lashes as shown in Fig.(1) [7].

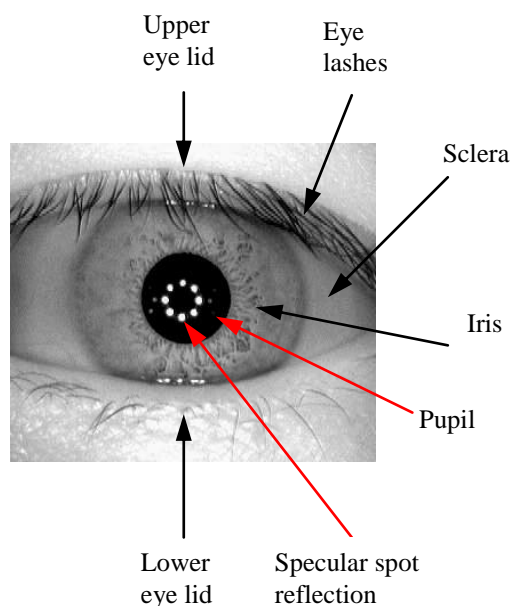


Fig.(1) An eye image.

The pupil is the darkest area in the eye image and it is best fit in a circle, pupil localization has many applications. Iris recognition has become more important in recent years and the accurate pupil detection, as the inner iris boundary, is the most

important and the first step in iris recognition system [1]. In particular, accurate estimation of the pupil size and centration is crucial in customized refractive surgery, corneal transplantation, and advanced contact lens fitting [2]. Numerous methods were proposed to locate the pupil based on the fact that the pupil edge can be estimated by circles, Daugman [3] had used integro-differential operator for pupil boundary detection, Wildes [4] had applied Hough transform in a binary edge point to estimate particular contour parameter values such as circle or ellipse. Many other researches have been conducted, they based on these two techniques [5].

Almost all of the earlier methods are sensitive to specular spot reflection and requires intensive computation.

2. The Proposed System

In this research work, eye images downloaded from the database belong to the Institute of Automation, Chinese Academy of Sciences (CASIA) [7] were used, and after analyzing these images, it has been found that the pupil takes about 10% of the entire eye image, and based on this observation, the threshold value was located in the eye image histogram by finding the gray level index where the accumulative number of pixels for that index is about 10% of the number of pixels in the entire eye image. The threshold value is then used to convert the eye image to binary

image (black and white) by applying binarization according to the threshold value, then the resulted eye image is segmented, and the largest black segment is selected to represent the initial allocation of pupil. The initial center point and radius of the pupil, could found for initial pupil are, after filling the pupil with black color to remove the specular spot reflection. The circle fitting algorithm was applied to find the exact pupil center and radius.

2.1 Finding Threshold

Finding the threshold value is the most important step, because the results of all the subsequent steps depend vitally on it. An erroneous threshold value will force the following steps to work either on a small portion of the pupil in the case of a low threshold value or on a large portion of the eye image including the pupil, lids, and lashes in the case of a high threshold value.

In this step, the histogram of the iris image is created first, then Algorithm (1) is used to

find the threshold value represented by the histogram index where the accumulative number of pixels is greater than or equal to 10% of the number of pixels in the eye image, Fig.(2-b) show a histogram for the eye image in Fig.(2-a) with the selected threshold value as the white line. Next, the image is converted to binary image (black and white) based on the determined threshold value as shown in Fig.(2-c).

Algorithm: Compute Threshold Value

Input: eye image.

Output: threshold value

```

pixel_co = 0
index = 0
max_size = eye image size / 10
while pixel_co < max_size
    pixel_co = pixel_co + histogram[index]
loop
return index

```

Algorithm (1): Finding the Threshold Value

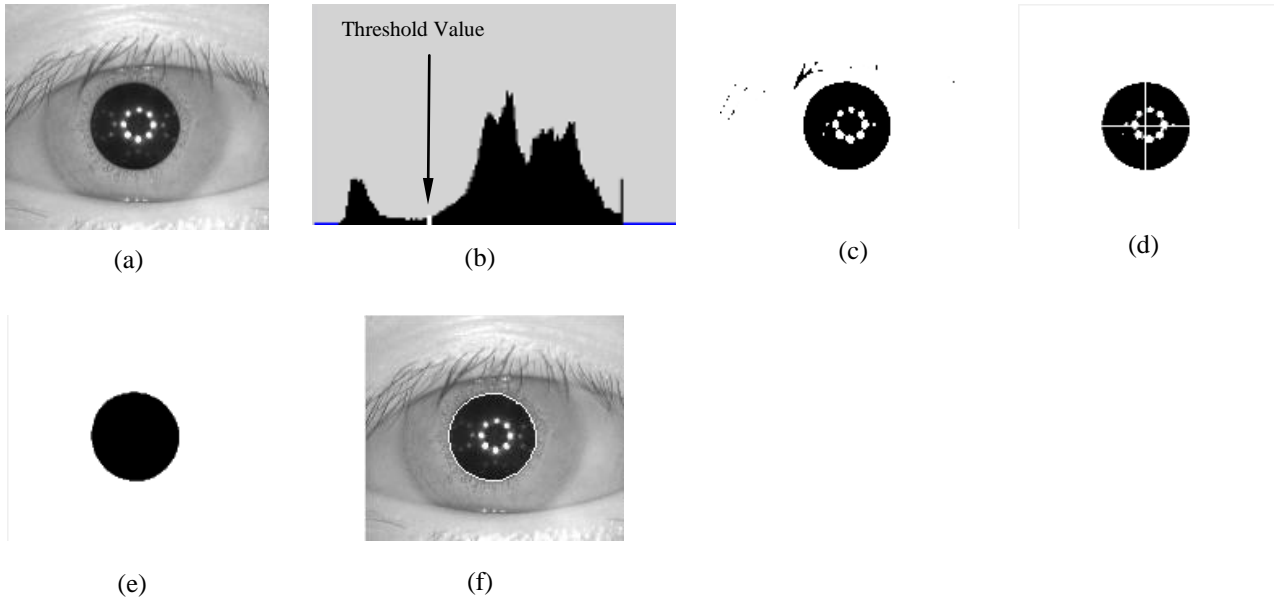


Fig.(2) Pupil localization steps: (a) original image, (b) histogram with threshold value in white, (c) binary image, (d) largest segment with computed pupil initial center and radius shown in white lines, (e) pupil segment filled with black color, (f) final image showing pupil boundary.

2.2 Segmentation

In this step, the resulted binary image is segmented using the seed fill algorithm. The steps of this algorithm are shown in Algorithm (4), the largest segment is computed using the Algorithm (3).

Seed fill algorithm uses array instead of recursion for faster performance. The segment points are registered in an array, cp is a pointer to the current pixel being processed in the segment array, lp is a pointer to the last pixel in the segment array. The algorithm checks the

color of the pixels to the right, left, down, and up of the current pixel, and it change any found black pixel to white and add it to the segment array.

The algorithm entitled "Find Pupil Area" is used to find the largest segment in the image by scanning the image pixels and call the seed fill algorithm on every black pixel to find the segment of all the connected pixels in which the black pixel belongs to, then it check if this segment is the largest segment in order to save their pixels coordinates in the *maxSegment* array, the algorithm will return the largest segment as shown in Fig.(2-d).

Algorithm: Seed Fill

Input: x and y coordinates of a pixel.

Output: change all connected pixels color from black (0) to white (255) starting from the given pixel and return the number of affected pixels.

```

segment(0).x = given x
segment(0).y = given y
cp = 0
lp = 0
while cp <= lp do
  x = segment(cp).x
  y = segment(cp).y
  if I(x+1,y) = 0 then
    I(x+1,y) = 255
    lp = lp + 1
    segment(lp).x = x+1
    segment(lp).y = y
  end if
  if I(x-1,y) = 0 then
    I(x-1,y) = 255
    lp = lp + 1
    segment(lp).x = x-1
    segment(lp).y = y
  end if
  if I(x,y+1) = 0 then
    I(x,y+1) = 255
    lp = lp + 1
    segment(lp).x = x
    segment(lp).y = y+1
  end if
  if I(x,y-1) = 0 then
    I(x,y-1) = 255
    lp = lp + 1
    segment(lp).x = x
    segment(lp).y = y-1
  end if
  cp = cp + 1
repeat
return lp

```

Algorithm (2): Seed fill algorithm

Algorithm: Find Pupil Area

Input: Image.

Output: maximum segment in the image.

```

width = width of the input image
height = height of the input image
mPixels = 0
for x = 1 to width do
  for y = 1 to height do
    if I(x,y) = 0 then
      sPixels = SeedFill(x,y)
      if sPixels > mPixels then
        mPixels = sPixels
        maxSegment = segment
      end if
    end if
  repeat
repeat
return maxSegment

```

Algorithm (3): Find Pupil Area algorithm

2.3 Calculation of Pupil Initial Center and Radius

After finding the pupil segment, the pupil initial center point (x_c , y_c) is computed from the following equations [6]:

$$x_c = \frac{1}{s} \sum_{c=0}^{s-1} p_x(c) \dots\dots\dots(1)$$

$$y_c = \frac{1}{s} \sum_{c=0}^{s-1} p_y(c) \dots\dots\dots(2)$$

Where x_c and y_c is the x and y coordinates of the pupil initial center point, p_x is the array of all pixel's x coordinates of the pupil segment, p_y is the array of all pixel's y coordinates of the pupil segment, and s is the number of pixels in the pupil segment.

Next the pupil initial radius is computed from the following equation [6]:

$$r = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2} \dots\dots\dots(3)$$

Where r is the pupil initial radius and (x_1 , y_1) is the first pixel in the circle as shown in Fig.(3).

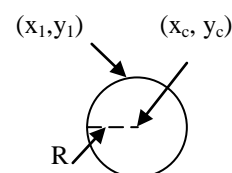


Fig.(3) Pupil initial center and radius.

Fig.(2-d) shows the pupil initial center point as the intersection of the two white lines which represents the pupil diameter (horizontally and vertically).

2.4 Filling the pupil

In order to remove the specular spot reflections, the pupil is filled with black color using Algorithm (4).

2.5 Fine Circular Fitting

As a last step in the proposed method, the determined pupil center point and radius coordinates are redetermined after removing the specular spot reflections and finding the best fitted circular mask on the pupil segment. This was accomplished by applying the steps of Algorithm (5).

In circle fitting algorithm, *isBlack* is a function return true if all pixels colors located on a pupil circle, with (Xc, Yc) as the center point coordinates and R as the radius, are black. This function returns true if the circle is entirely inside the pupil segment were all pixels are black. The algorithm try to fit a circle in the pupil segment starting by the circle with the initial pupil center point and radius, then increase the radius and checks if the circle still inside the pupil and if it is not it tries to move the circle to the left, right, up, down, and to the other four diagonal directions, if any of them work, the algorithm will continue to increase the radius and if it is not then the algorithm will stop, the result of this step are shown in Fig.(2-f).

Algorithm: Pupil Fill

Input: Pupil segment.

Output: Filled pupil segment.

```
height = Height of pupil segment
width = Width of pupil segment
hw = width / 2
hh = height / 2
for y = 1 to height do
  flag = false
  for x = 1 to hw do
    if P(x,y) = 0 then
      flag = true
```

```
    else if flag = true
      P(x,y) = 0
    end if
  repeat
    flag = false
    for x = width down to hw+1 do
      if P(x,y) = 0 then
        flag = true
      else if flag = true
        P(x,y) = 0
      end if
    repeat
  repeat
for x = 1 to width do
  flag = false
  for y = 1 to hh do
    if P(x,y) = 0 then
      flag = true
    else if flag = true
      P(x,y) = 0
    end if
  repeat
  flag = false
  for y = height down to hh+1 do
    if P(x,y) = 0 then
      flag = true
    else if flag = true
      P(x,y) = 0
    end if
  repeat
  repeat
```

Algorithm (4): Pupil filling algorithm

Algorithm: Circle Fitting

Input: Pupil segment.

Output: Pupil center and radius

isBlack(Xc,Yc,R) : retrun true if all the pixels colors located on pupil circle with (Xc,Yc) center and R radius are black

```
Xc = initial pupil x center
Yc = initial pupil y center
R = initial pupil radius
finish = false
while finish = false do
  R = R + 1
  while isBlack(Xc,Yc,R)=true do
    R = R + 1
  repeat
  if isBlack(Xc+1,Yc,R)=true then
    Xc = Xc + 1
  else if isBlack(Xc-1,Yc,R)=true then
    Xc = Xc - 1
  else if isBlack(Xc,Yc+1,R)=true then
    Yc = Yc + 1
  else if isBlack(Xc,Yc-1,R)=true then
    Yc = Yc - 1
```

```

else if isBlack(Xc-1,Yc-1,R)=true then
    Xc = Xc - 1
    Yc = Yc - 1
else if isBlack(Xc+1,Yc-1,R)=true then
    Xc = Xc + 1
    Yc = Yc - 1
else if isBlack(Xc+1,Yc+1,R)=true then
    Xc = Xc + 1
    Yc = Yc + 1
else if isBlack(Xc-1,Yc+1,R)=true then
    Xc = Xc - 1
    Yc = Yc + 1
else
    finish = true
end if
repeat

```

Algorithm (5): Circle Fitting algorithm

3. Test Results

The proposed system was tested on ten eye images downloaded from the Institute of Automation, Chinese Academy of Sciences CASIA-IRISV3-Interval database.

For each image, the actual number of pixels occupied by the pupil was counted semi-manually by filling the pupil area with red color using Microsoft Windows Vista Paint program, then the number of red pixels was counted using a program that scans the iris image looking for red pixels. The actual pupil center was computed using equations 1 and 2 and the actual pupil radius was computed using equation 3.

Then, for each image the pupil pixels was counted using the proposed system by localizing the pupil, fill the occupied region with red pixels, count them, and compute the

pupil center and radius using equations 1, 2, and 3.

The actual pupil area in pixels and the detected pupil area error ratio are shown in Table (1) for thresholds 2%, 4%, 6%, 8%, 10%, 12%, 14%, 16%, and 18% of the image size. The error was counted using the following equation:

$$A_a = \frac{S_d - S_r}{S_r} \times 100\% \quad (4)$$

Where S_r is the real pupil size and S_d is the detected pupil size.

Table (2) shows the actual pupil center position (X_c , Y_c) in pixels and the detected pupil center position error versus threshold values. The position distance error was computed using the following equation:

$$D_e = \sqrt{(X_r - X_d)^2 + (Y_r - Y_d)^2} \quad (5)$$

Where (X_r , Y_r) is the real pupil center point position and (X_d , Y_d) is the detected pupil center point position.

Table (3) shows the actual pupil radius in pixels and the detected pupil radius error ratio versus the threshold values. The radius error was computed using the following equation:

$$A_r = \frac{R_d - R_r}{R_r} \times 100\% \quad (6)$$

Where R_r is the real radius and R_d is the detected radius.

Fig.(4) shows the results of applying the proposed method on image number 8.

Table (1)
Actual pupil area in pixels and detected pupil area error ratio.

#	Actual pupil area	Pupil area error ratio								
		using 2% threshold	using 4% threshold	using 6% threshold	using 8% threshold	using 10% threshold	using 12% threshold	using 14% threshold	using 16% threshold	using 18% threshold
1	6060	-88.2%	-74.1%	-4.8%	-0.1%	-0.1%	-0.1%	+4.6%	+4.6%	+9.2%
2	4762	-88.7%	-10.2%	-5.5%	-0.6%	-0.6%	-0.6%	-0.6%	+5.1%	+17.4%
3	5778	-90.8%	-79.7%	-4.7%	-0.1%	-0.1%	-0.1%	+4.7%	+4.7%	+19.6%
4	4754	-85.8%	-10.1%	-5.3%	-0.4%	-0.4%	-0.4%	-0.4%	-51.3%	-54.6%
5	4523	-82.3%	-10.5%	-0.5%	-0.5%	-0.5%	-0.5%	-0.5%	+4.7%	+4.7%
6	4503	-74.7%	-10.1%	-5.1%	-5.1%	+0.0%	+0.0%	-53.5%	-31.0%	+66.6%
7	5274	-83.0%	-14.6%	-5.1%	-5.1%	-0.4%	+9.4%	+14.8%	+20.2%	+31.6%
8	6706	-89.4%	-81.4%	-9.8%	-5.5%	-1.3%	+3.0%	-13.3%	-4.5%	-8.0%
9	5133	-75.8%	-21.1%	-2.5%	+2.3%	+2.3%	+2.3%	+12.4%	+66.5%	+92.0%
10	5017	-85.3%	-10.3%	-5.6%	-0.3%	-0.3%	+4.7%	+4.7%	+16.8%	+22.4%
Average error		84.4%	32.2%	4.9%	2.0%	0.6%	2.1%	10.9%	20.9%	32.6%

Table (2)
Actual pupil center position in pixels and detected pupil center position error.

#	Actual pupil center	Pupil center position error								
		using 2% threshold	using 4% threshold	using 6% threshold	using 8% threshold	Using 10% threshold	using 12% threshold	using 14% threshold	using 16% threshold	using 18% threshold
1	(167,135)	28.0	21.4	0.0	0.0	0.0	0.0	1.0	1.0	1.4
2	(155,170)	24.0	1.0	1.0	1.4	0.0	0.0	0.0	0.0	71.3
3	(164,178)	30.0	24.1	0.0	0.0	0.0	0.0	1.0	0.0	28.1
4	(157,165)	25.5	2.2	1.0	1.0	1.0	1.0	1.0	115.3	111.3
5	(158,174)	22.0	1.4	1.0	0.0	1.0	1.0	1.4	1.4	1.4
6	(152,143)	17.1	1.4	1.0	1.0	0.0	0.0	113.5	106.5	77.5
7	(168,144)	23.1	1.4	0.0	0.0	1.0	2.0	2.0	2.0	3.2
8	(158,166)	31.1	26.0	1.0	0.0	1.0	15.0	28.9	31.6	42.4
9	(146,132)	18.0	3.2	0.0	0.0	1.0	1.0	2.0	39.4	37.1
10	(169,145)	24.1	1.4	0.0	0.0	0.0	0.0	0.0	61.5	57.8
Average error		24.29	8.35	0.5	0.34	0.5	2	15.08	35.87	43.15

Table (3)
Actual pupil radius in pixels and detected pupil radius error ratio.

#	Actual pupil radius	Pupil radius error ratio								
		using 2% threshold	using 4% threshold	using 6% threshold	using 8% threshold	using 10% threshold	using 12% threshold	using 14% threshold	using 16% threshold	using 18% threshold
1	44	-65.9%	-50.0%	-2.3%	+0.0%	+0.0%	+0.0%	+2.3%	+2.3%	+4.5%
2	39	-66.7%	-5.1%	-2.6%	+0.0%	+0.0%	+0.0%	+0.0%	+2.6%	+7.7%
3	44	-70.5%	-56.8%	-4.5%	-2.3%	-2.3%	-2.3%	+0.0%	+0.0%	+6.8%
4	40	-65.0%	-7.5%	-5.0%	-2.5%	-2.5%	-2.5%	-2.5%	-32.5%	-35.0%
5	40	-60.0%	-10.0%	-5.0%	-5.0%	-5.0%	-5.0%	-5.0%	-2.5%	-2.5%
6	38	-50.0%	-5.3%	-2.6%	-2.6%	+0.0%	+0.0%	-60.5%	-42.1%	+5.3%
7	42	-59.5%	-9.5%	-4.8%	-4.8%	-2.4%	+2.4%	+4.8%	+7.1%	+11.9%
8	48	-68.8%	-58.3%	-8.3%	-6.3%	-4.2%	-2.1%	-10.4%	-6.3%	-8.3%
9	39	-48.7%	-7.7%	+2.6%	+5.1%	+5.1%	+5.1%	+10.3%	+33.3%	+43.6%
10	40	-62.5%	-5.0%	-2.5%	+0.0%	+0.0%	+2.5%	+2.5%	+7.5%	+10.0%
Average error		61.8%	21.5%	4.0%	2.9%	2.1%	2.2%	9.8%	13.6%	13.6%

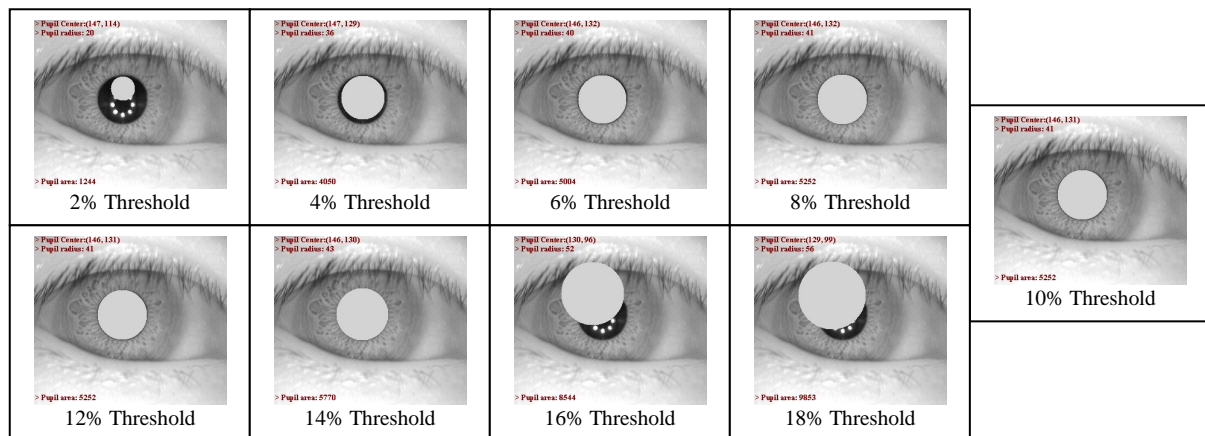


Fig.(4) Images-8 results with circles filled with red showing the pupil area for different thresholds.

4. Conclusions

In this paper, a simple, fast and efficient algorithms are proposed for locating the pupil center point and radius based on a threshold value of 10% of the image size instead of the complex computations being used by many other researches.

Test results indicate that the best threshold value lies between 6% and 12% of the image size. A threshold value of 10% of the image size leads the subsequent steps to detect the pupil area and pupil radius with a minimal error ratio as indicated by Table (1) and Table (3). A 8% of the image size threshold value leads the subsequent steps to detect the pupil center position with a minimal error ratio as indicated by Table (2).

A 8% of the image size threshold value detect a small amount of the pupil area which leads to incorrect localization, while a 10% of the image size threshold value detect the most accurate pupil area and pupil radius with a very minimum error ratio in pupil center position which leads to accurate pupil localization.

Reference

- [1] Rasoul Kheirolahy, Hossein Ebrahimnezhad, and Mohammad Hossein Sedaaghi, "Robust Pupil Boundary Detection by Optimized color Mapping for Iris Recognition", CSICC 14th International CSI, 2009, pp.170-175.
- [2] D. Robert Iskander, Michael J. Collins, Siegfried Mioschek, and Martin Trunk, "Automatic Pupillometry From Digital Images", Biomedical Engineering, IEEE Transactions on, Vol. 51, NO 9, 2004, pp. 1619 - 1627.
- [3] J. Daugman, "How iris recognition works", IEEE Trans. CSVT, Vol. 14, NO. 1, 2004, pp. 21 - 30.
- [4] R. Wildes, "Iris recognition: An emerging biometric technology", World Scientific and Engineering Academy and Society (WSEAS), Vol. 85, NO 9, 2007, pp. 91-96.
- [5] Imene Khanfir Kallel, Dorra Sellami Masmoudi, Nabil Derbel, "Fast Pupil Location For Better Iris Detection", Systems, Signals and Devices, SSD '09, 6th International Multi-Conference on, 2009, pp. 1-6.
- [6] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson Education, 2008 Third Edition.

- [7] Center for Biometrics and Security Research, CASIA iris image database <http://www.cbsr.ia.ac.cn/irisdatabase>.

الخلاصة

في هذا البحث، تم اقتراح طريقة بسيطة وسريعة، ودقيقة لتحديد البؤبؤ. بعد تحليل عدد من صور العين وجدوا ترددات الوانها تم ملاحظة ان منطقة البؤبؤ تشكل نسبة 10% من المساحة الكلية لصورة العين وتمثل اكثر منطقة سواداً. تم حساب قيمة حدية من جدول ترددات الالوان لكي يستخدم لتحويل الصورة الى صورة ثنائية، وتم استخدام خوارزمية ملئ البذرة لتقسيم الصورة الى مقاطع وايجاد مقطع البؤبؤ و الذي يمثل اكبر مقطع، ثم ملئه باللون الاسود لازالة اي انعكاسات ضوئية. وتم تحديد دائرة البؤبؤ باستخدام خوارزمية الملائمة الدائرية. بعد اجراء الاختبارات على 10 صور منزلة من قاعدة بيانات " CASIA-IRISV3-Interval" تم التوصل الى ان متوسط نسبة الخطأ في مساحة البؤبؤ هي 0.6%، متوسط الخطأ في نقطة المركز للبؤبؤ هي 0.5%، ونسبة متوسط الخطأ في نصف قطر البؤبؤ هي 2.1%.