Iris Segmentation using Geometrical Methods

Gustavo L. Vidal-González[1], Alejandro A. Ramírez-Acosta[2],
Mireya S. García-Vázquez[1], and Ciro A. Martínez García-Moreno[1]

[1] Centro de Investigación y Desarollo de Tecnología Digital - CITEDI,
Av. del Parque No. 1310, Mesa de Otay,
Tijuana, Baja California, 22510,
Mexico
[2] MIRAL R&D, Imperial Beach,
USA
{vidal, mgarcia, ciro}@citedi.mx, ramacos10@hotmail.com

Abstract. In the field of biometric identification, iris recognition systems employ techniques to isolate the iris region to speed up the following stage of identification. Many existing iris segmentation techniques rely heavily on image operations to obtain high levels of accuracy but introduce unneeded computational overhead. Others offset the wide use of image operations by analyzing the image prior to attempting segmentation while maintaining a good deal of accuracy. Fewer still, offset the use of image operations even further by adopting fast deterministic approaches to the iris segmentation problem based on prior knowledge of the geometry of the iris, but achieve only modest accuracy. We propose a method for iris segmentation that focuses on minimum computational overhead while maintaining a good deal of accuracy which utilizes: pre-segmentation techniques based on image histogram analysis, canny edge detection image operations and finally a geometrical analysis to segment the iris region.

Keywords: Iris Localization, Circle Detection, Canny Edge Detection, Histogram Analysis, Iris Biometrics.

1 Introduction

The use of biometric identification systems has extended rapidly, due to the need of security systems that allow entry to access points based on individual verification. It is well known that biometric identifiers are of great use for individual human recognition. Thus, individual features such as fingerprints, iris textures, voice and etcetera are used as some of the more popular biometric identifiers for these security verification systems.

These related technological advances have been of great help for biometric identification because security systems employ a complex infrastructure that integrates a great deal of hardware as well as software to accomplish the task of identification. The present work utilizes identification through iris biometrics; it is worth mentioning that our area of interest is in the software of biometric identification. It is based on two related previous works that have applied quick

Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 73-84

deterministic approaches [1] (Liu *et al.*) and more accurate thorough approaches [2] (Sun *et al*.) to iris segmentation. This article presents a method of improvement in the detection of the iris by leveraging part of the accuracy of Sun *et al*. [2] using pre-segmentation techniques through histogram analysis and then proceeding with quick iris segmentation methods proposed by Liu *et al* [1]. This allows us to work with a standard set of grayscale iris images like the "NIR still eye image" set from the MBGC (Multi-Biometric Grand Challenge) [3].

The method defined in this article uses a pupil pre-segmentation technique by means of a threshold obtained from a histogram analysis of the original image [2]; Canny Edge Detection [4], to obtain pupil/iris contours; Traditional geometric methods to obtain the pupil center and pupil/iris radius [1]; and finally an outward scan starting from the pupil/iris radius to find the iris/sclera boundary [1].

This paper is organized as follows, in section 2 we discuss pre-segmentation through histogram analysis, in section 3 we show the result of obtaining contours using Canny edge detection, in section 4 we locate the pupil center as well as the pupil/iris boundary using geometrical methods, the modeling of a parametric circular motion equation is explained for the purpose of locating the iris/sclera boundary in section 5, in section 6 we discuss the results of the iris segmentation and finally we bring to light some conclusions of using this method in section 7.

2 Pre-segmentation Based on Histogram Analysis

Geometrical methods rely on obtaining genuine circle contours along the pupil/iris boundary after applying Canny edge detection. However, when the eye is not delimited in the image, more contours may be detected, be it from eye lashes, eyelids, eye brows, wrinkles, scars and etcetera. Therefore steps should be taken to facilitate the localization of the pupil/iris boundary. For example: an image acquisition system where the user's eye closes-in into the camera with appropriate lighting [1], thus producing an image that is easy to work with; the method explained in this article assumes a subpar image with eyelashes, eyelids and eyebrows included in the image.

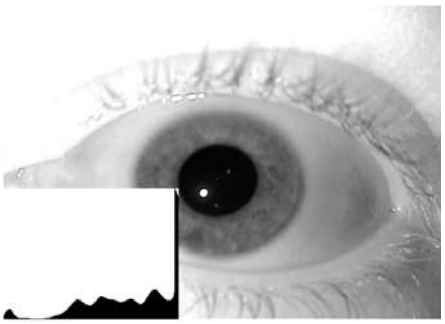

Fig. 1. Iris image with super-imposed Intensity histogram (*H*).

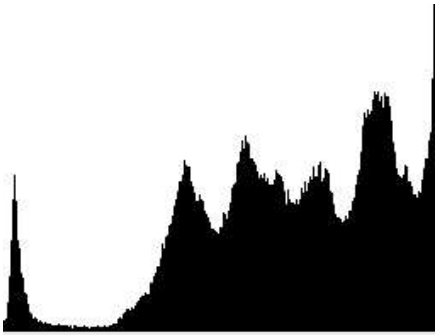

Fig. 2. Close up of Intensity histogram (*H*).

Another technique for delimiting the pupil region is applying a dark intensity threshold based on a histogram analysis. Being that the darkest grayscale intensities

are mostly located in the region of the pupil, the threshold provides coarse pupil segmentation therefore eliminating many unwanted objects in that image [2].

Given that the threshold value should be carefully considered, we propose a method consisting of low pass filtering and derivative like operations on the histogram to faithfully locate a good threshold value from the histogram.

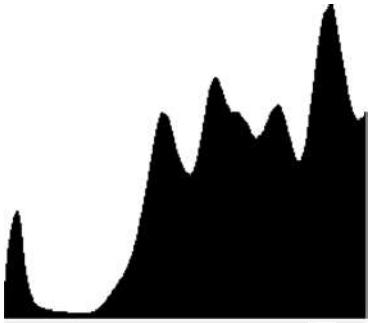

Fig. 3. Average filtered histogram ($\bar{H}$).

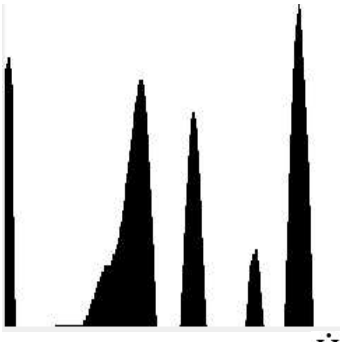

Fig. 4. Pseudo-derived histogram ($\dot{H}$)

The detailed method for obtaining the appropriate threshold value is described as follows:

1) Pixels of image are read in standard 8 bit grayscale format. The histogram (*H*) is calculated by surveying all the grayscale pixel intensities in the iris image and storing the occurrence of the intensities in 256 (2^8) possible bins. If quantization of intensities by means of bin reduction (i.e. 128, 64, 32 bins) would be used it would hinder the ability of the process to pinpoint a needed single grayscale intensity/bin. When the image histogram is plotted the darkest intensities are the left most values in the histogram corresponding to the near-zero most values and the brightest intensities are the right most values in the histogram (as seen in figure 1) (only positive values have been plotted).

2) Upon closer inspection of the raw histogram (H) (see figure 2), trends in the form of valleys and peaks could be seen with each one having their own frequency. However, if a valley or peak detection routine was to be applied to the raw histogram, an incorrect point maybe selected due to a high frequency spike (it is difficult to determine with certainty where overall peaks and valleys may occur).

To avoid this problem, an average filter is applied to the histogram to smooth out any high frequency spike. At the same time, the average filter helps lower the level of the most repeated intensities thus resulting in a more level balanced average filtered histogram ($\bar{H}$) where lower frequency trends are more visible and the histogram is smoothed out (see figure 3). Equation 1 is used to apply average filtering to the raw histogram (*H*) where (N_f) is the order of the filter *(N_f >3 and must be odd number)* to be applied, reference index (*i=0, 1, 2, …, 255*) will be used for the calculation of all resulting ($\bar{H}$) elements and supporting index (*j*) will point to the (*H*) elements needed to be added to the accumulator prior to dividing it by the *(N_f)* number of elements.

$$\bar{H}_t = \frac{1}{N_f} \sum_{j=t-(N_f-1)/2}^{t+(N_f-1)/2} H_j \tag{1}$$

For simplicity, *Hj=0* when (*j*) index is out of bounds (*j<0 or j>255*).

3) We interpret the resulting ($\bar{H}$) to be a discrete one dimensional signal so we can apply a pseudo-derivative ($\dot{H}$) operation (equation 2) to detect maximums and minimums in ($\bar{H}$). The pseudo-derivative differs from the discrete derivative in $\Delta x=N_f -1$ ($\Delta x=1$ *for discrete derivative)* and the elements evaluated are equally spaced near left and near right neighboring elements of the discrete signal ($\bar{H}$) (*in discrete derivative, elements are adjacent*). The same value of (N_f) used in equation 1 is used for equation 2. By using the pseudo-derivative small changes sudden changes in ($\bar{H}$) are attenuated whilst considerable changes in ($\bar{H}$) become more appreciable.

$$\dot{H}_t \approx \frac{\bar{H}_{t+(N_f-1)/2} - \bar{H}_{t-(N_f-1)/2}}{N_f - 1} \tag{2}$$

The indexing element (*i = 0, 1, 2, ..., 255*) is used in the calculation of all elements of ($\dot{H}$) (see figure 4), also all out of bounds indexed elements of ($\bar{H}$) are considered to be zero. The value of N_f used was 13 which netted good results during experimentation.

4) Finally a routine is used to detect changes in polarity of the slopes starting from $\dot{H}_0$ *to* $\dot{H}_{255}$. A change from positive to negative is stored as a peak and a change from negative to positive is stored as a valley.

5) The location (or index) of the first considerable valley in the histogram is taken as the intensity threshold [2]. (See resulting effect applying dark intensity threshold in iris image in figure 5).

It is noteworthy to state that no further segmentation will be applied to remove occluding eyelashes, which may still be present after the dark threshold applied to the iris image.

Fig. 5. Coarse pupil localization obtained by applying intensity threshold to iris image.

3 Canny Edge Detection

Within the algorithms available for contour detection in image processing, we use a proven method known as Canny edge detection [4]. Canny edge detection employs a combination of low pass image filtering, followed by high pass filtering to obtain horizontal and vertical gradient component images. The contour image (see figure 6) formed by the magnitude of the component images is further thinned out through a phase/angle analysis.

Once Canny edge detection has been applied to the pre-segmented image (figure 5), the process of collecting pupil/iris contour points can be started. The pixel coordinate collection process used is horizontal row scanning, approaching from the left as well as approaching from the right. The approaching scan from either direction is stopped where any appreciable pixel is detected. Thus, detection of only two or less pixels per horizontal row in the image is possible. Pixels detected have their coordinates stored in a 2D array. Even though it is possible that points belonging to an anomaly such as light reflection may be selected, these will not conform to the constraints of the next process of pupil center localization and as such be subsequently discarded.

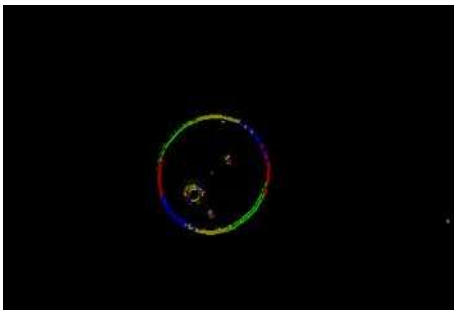

Fig. 6. The resulting image after Canny edge detection.

4 Pupil Center and Pupil/Iris Boundary Localization using Geometrical Methods

The method for pupil center localization parts from the simple geometry problem of formulating a circle equation from three points.

The summarized solution is as follows: Given three points in a 2D plane (figure 7a), midpoints between the original points are calculated (figure 7b). Two new line equations are formulated in such way that they pass through the calculated midpoints and are perpendicular to the previous lines. The intersection point of these new lines is considered to be the circle center (figure 7c). The radius can be obtained from calculating the distance from the circle center to any of the original three points (figure 7d). Finally an appropriate circle equation can be formulated parting from a known circle center and a known radius (figure 7d). The detailed process is explained fairly well by [5].

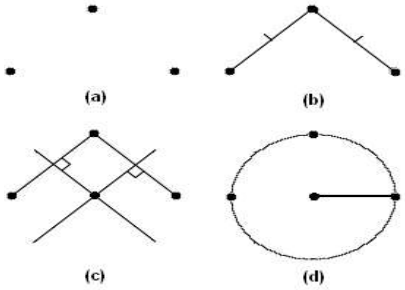

Fig. 7. Formulating circle equation, given 3 points in 2D plane.

While all most any set of three points in a 2D plane can be used to formulate a circle. This is less true for four points, as an implicit constraint arises where these points have to belong to a circular type of object for them to have one conceding intersection point from their derived intersection lines [1]. The constraint is even more severe, when more than four points are taken into account.

The method described in this paper selects four arbitrary points that are evenly distributed, as they should be points that are representative of the remaining contours. These must be selected from the 2D array where pixel coordinates where stored after Canny Edge Detection. Midpoints which contain the necessary data to formulate the intersecting lines [5] are then calculated.

We analyze the slope angle (θ) of the intersection lines prior to formulating them in rectangular format. By utilizing the "arctan2" function, in which computer compilers handle the numerator (m_y) and denominator (m_x) of the slope (m) as separate parameters, division by zero can be avoided (see equation 3). With the use of the "arctan2" function, (θ) will yield values ranging from ($-\pi$ to π).

$$\theta = arctan2(m_y, m_x) \tag{3}$$

To simplify the interpretation of the slope angle (θ), we add (π) to a negative result (as seen in equation 4).

$$\theta = \begin{cases} \theta + \pi_i & \theta < 0 \\ \theta_i & \theta \geq 0 \end{cases} \tag{4}$$

Based on the slope angle (θ) (see equation 5) we formulate the linear equations, firstly in the slope intercept form ($y=mx+b$ *for near horizontal lines or* $x=my+b$ *for near vertical lines*), where (b) is the intercept along the appropriate axis [5]. Then secondly in the standard form ($k_1x+k_2y=k_3$), which is achieved by isolating (b), where (k_1, k_2 *and* k_3) are the known constants.

Once we have formulated at least three intersection lines in standard form (see left side of equation 5) we can proceed to calculate an intersection point for all lines solving the set of linear equations using Kramer determinants.

$$k_1x + k_2y = k_3 \leftarrow \begin{cases} y = m_1x + b_1, & \theta \leq \frac{\pi}{4} \\ x = m_2y + b_2, & \frac{\pi}{4} < \theta < \frac{3\pi}{4} \\ y = m_3x + b_3, & \theta \leq \frac{3\pi}{4} \end{cases} \tag{5}$$

In the case where we have only one solution point, for the moment we consider this to be the pupil center and proceed to obtain the pupil/iris radius by calculating the distance to any of four originally selected contour points. Finally we measure the number of pixels in the contour image found along a circle perimeter (with the calculated pupil center and pupil radius) to ascertain the results.

In the case where there is no single solution point for the intersecting lines, four new contour points are selected until a single solution point for the new intersecting lines is established.

5 Iris/Sclera Boundary Localization

As stated previously, this paper assumes that two concentric circles that define the pupil/iris boundary and the iris/sclera boundary are centered on a single point [1]. The iris/sclera boundary will be detected by identifying a reasonable contrast change in data collected from an outwards scan parting from the pupil/iris boundary. In practice, most iris images contain anomalies such as light reflection which can be misinterpreted as a contrast change when scanning for the iris/sclera boundary. Or the regions containing the eyelids or eyelashes may also be misinterpreted as a significant contrast change. For small anomalies located in the iris region, the data collected from problem pixel intensities can be average filtered with like pixels in the iris that share the same radius (or distance) from the pupil center.

5.1 Pixels Scan using Circular Motion

A model for obtaining the pixel indexes with the intention of scanning pixels along a circle perimeter within the image is needed. The model devised is based on the formula for parametric circular motion shown with equation 6.

$$P(k) = (r \cdot \cos\omega \tag{6}$$

Where (r) is the radius of the circle, (ω) is the angular frequency of the circular motion, (x_c) and (y_c) are the horizontal and vertical displacement of the circle center (for our application the index of the located pupil center) and (k) is the function variable.

Parting from the relationship of angular frequency (ω) and period (T) as shown in equation 7 where (T) is the perimeter of the circle for this application of scanning along the circle perimeter in one cycle, (ω) can be simplified in terms of (r).

$$\omega = \frac{2\pi}{T} = \frac{2\pi}{2\pi r} = \frac{1}{r} \tag{7}$$

Thus equation 6 and equation 7 can be combined for expressing (ω) in terms of (r) for this application as shown in equation 8:

$$P(k) = \left(r\cos\frac{k}{r} + x_c, r\sin\frac{k}{r} + y_c\right) \tag{8}$$

5.2 Mean Pixel Intensity of a Circle Perimeter Scan

Given that the pupil center in the image has been located, and that the distance from this point to the pupil/iris boundary is known. This will be our initial radius (r_i) for scanning outwards in a circular motion. After the mean pixel intensity data ($\bar{C}_r$) has been collected and stored for this radius (r_i), the radius will be increased by one pixel and new data will be collected for (r_{i+1}) this data collection process will continue with increasing radii until an established final radius (r_f) is reached or the scan yields an out of bounds index in the image.

The following equation 9, is an example of obtaining the mean pixel intensities ($\bar{C}_r$) in the image (Img) that were selected by indexing in a circular motion. The perimeter comes into play in the calculation of the mean as its value contains the number of pixel intensities that were scanned and as a stopping limit for the parametric variable (k).

$$\bar{C}_r = \frac{1}{2\pi r}\sum_{k=0}^{2\pi r-1} Img\left[r\cos\left(\frac{k}{r}\right) + x_c, r\sin\left(\frac{k}{r}\right) + y_c\right] \tag{9}$$

In addition, the model must also have the capacity to scan only certain segments of the circle perimeter, delimited by fixed constraint angles with respect to the pupil center. In this manner, a good deal of pixel information regarding eyelashes and eyelids is omitted. The ones proposed in the related work by [1] are used: From -π/4 to π/4 and from 3π/4 to 5π/4 (see figure 8 for a projection of the delimited segments).

These delimiting angles can be translated to a corresponding (k) variable value (being 2πr one complete cycle): From -πr/4 to πr/4 and from 3πr/4 to 5πr/4.

Taking these new segmented circle restrictions and applying them to equation 9 yields the delimited equation 10.

$$\bar{C}_r = \frac{1}{\pi r}\sum_{k=\frac{-\pi r}{4}}^{\frac{\pi r}{4}} Img\left[r\cos\left(\frac{k}{r}\right) + x_c, r\sin\left(\frac{k}{r}\right) + y_c\right] + \frac{1}{\pi r}\sum_{k=\frac{3\pi r}{4}}^{\frac{5\pi r}{4}} Img\left[r\cos\left(\frac{k}{r}\right) + x_c, r\sin\left(\frac{k}{r}\right) + y_c\right] \tag{10}$$

In many iris images, the iris/sclera boundary is not well defined and usually presents a low contrast change. When analyzing the data ($\bar{C}_{r_s}, \bar{C}_{r_s+1}, \ldots, \bar{C}_{r_f}$) it is difficult to establish with certainty which radius represents the iris/sclera boundary. For this reason, consecutive circle scans are grouped into layers representing the average pixel intensity conformed of several circle scans. The layer size (L_{size}) value chosen during the experimentation was three (the same as used in [1]), which netted good results for the "NIR still eye image" set [3].

The mean layer intensity ($\bar{L}$) (below in equation 11) can be used to represent a group of consecutive ($\bar{C}_r$) data from a starting radius (r_s) to the ending radius (r_s+ L_{size}-1) within the layer.

$$\bar{L} = \frac{1}{L_{size}} \sum_{r=r_s}^{r_s+L_{size}-1} \bar{C}_r \tag{11}$$

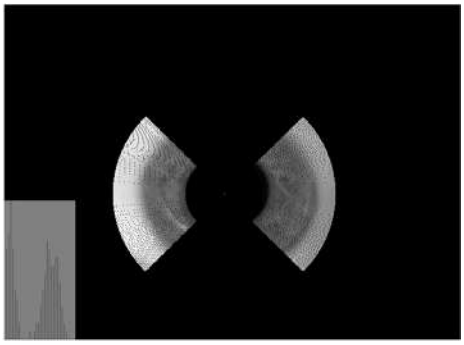

Fig. 8. Outwards scan to find the largest contrast change parting from pupil/iris boundary (lower left plot: mean layer intensity vs. layer radius).

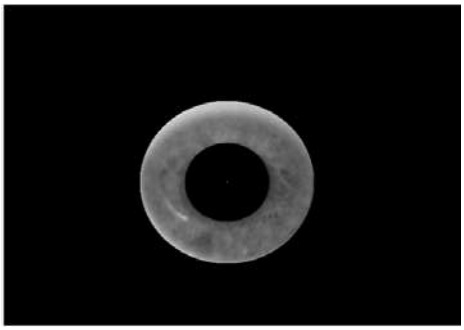

Fig. 9. Segmented iris image of figure 1.

Thus by analyzing the consecutive mean layer intensities ($\bar{L}$), we can find the highest contrast change from one layer to another (the first few layers will be ignored as some of the dark pixels of the pupil may intervene in the first layers, see figure 8).

The layer radius in which the high contrast change occurs will be recorded as the radius that defines the iris/sclera boundary. The final segmented image for figure 1 is seen in figure 9.

6 Results/Segmentation

Results very similar to the ones found by the method proposed by Liu *et al* [1] where obtained. Even in some images with occluded irises, the pupil center point, iris/pupil boundary, and iris/sclera boundary where genuinely located by the method (see figure 10 and figure 11).

Fig. 10. Original iris image.

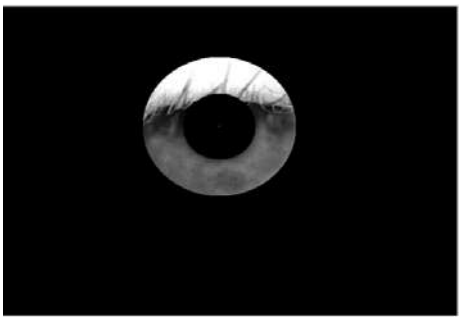

Fig. 11. Segmented iris image.

The geometrical method relies in many ways on the localization of the pupil center. As stated, genuine circle points must be selected to construct appropriate lines for the pupil center point localization. Thus problems may arise when there are too many contours left after canny edge detection (see figure 12, figure 13 and figure 14).

Fig. 12. Original iris image.

Fig. 13. Eyelash contours remain after Canny edge detection.

This would affect the next stage where non-genuine circle points would be collected during the circle contour point collecting routine (effect seen in figure 14).

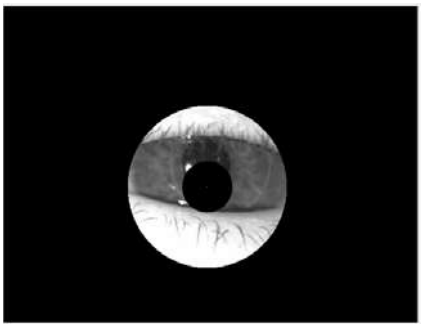

Fig. 14. Failed Iris Segmentation.

7 Conclusions

Parting from the assumption that iris images taken from a controlled iris image set contain considerably well defined Euclidean circles, the method proposed provides good iris localization with less computational latency by reducing image operations and dropping Hough circle detection on the iris image. The method proposed suffers when iris images of very low resolution are used (accuracy problems) and when there are far too many anomalies in the iris image (after canny edge detection).

7.1 Comparison with Previous Methods

Practical comparison with the previous methods mentioned in this article is not easily achieved. For comparison with the method proposed by Liu *et al* [1], while the results are visually similar for iris segmentation, in our proposed method no eye image acquisition system is being used (fundamental component of method [1]). As for comparison with the method proposed by Sun *et al* [2], while we share the similar component of histogram analysis, the focus of our method was always intended to reduce computational demanding operations like image morphology or Hough circle detection present in method [2]. Furthermore ascertaining a quantifiable measurement between the three methods for comparison is difficult, since there is no standardized measurement for iris segmentation.

It is also to be noted that state of the art methods focus on the solution to the iris biometric identification problem as a whole; however, iris segmentation does not conform the total solution in itself, but as a component can still be very useful.

Acknowledgments. This work was supported by IPN-SIP20100030.

References

1. Liu, Y., Yuan, S., Zhu, X., Cu, Q.: A practical Iris Acquisition System and A Fast Edges Locating Algorithm in Iris Recognition. In: IMTC 2003-Instrumentation and Measurement Technology Conference, Vail, CO, USA, pp. 20-22 (2003)
2. Sun, C., Zhou, C., Liang, Y., Liu X.: Study and Improvement of Iris Location Algorithm. In: ICB 2006, LNCS 2832, pp.436-442 (2005)
3. NIST Multiple Biometric Grand Challenge. http://face.nist.gov/mbgc/
4. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 8 (6):679–698 (1986)
5. Bourke, P.: Equation of a circle from 3 points (2 dimensions), http://local.wasp.uwa.edu.au/~pbourke/geometry/circlefrom3/