Iris Localization via Pulling and Pushing

Zhaofeng He, Tieniu Tan, and Zhenan Sun
Center for Biometrics and Security Research & National Laboratory of Pattern Recognition,
Institute of Automation,Chinese Academy of Sciences
{zfhe, tnt, znsun}@nlpr.ia.ac.cn

Abstract

Iris localization is a critical module in iris recognition because it defines the inner and outer boundaries of iris region used for feature analysis. State-of-the-art iris localization methods need to implement a brute-force search of the large parameter space, which is time-consuming and sensitive to noises. This paper proposes a novel iris localization method based on a spring force-driven iteration scheme. First, the coarse localization of pupil is obtained by an AdaBoost-based iris detection method. Then the radial edge points are detected in polar coordinate, which contribute a force on the circle center based on Hooke's law. Finally the center and radius of pupil and iris are refined according to the composition of forces from all points. After 3-4 iterations, the precise location of iris could be found. Experimental results show that our method is faster and more accurate than state-of-the-art iris localization methods.

1. Introduction

With the increasing demands of security in our daily life, personal identification based on biometrics has become an emerging alternative for security applications [1]. Among the biometric family, iris is particularly useful for its unique, stable, non-invasive features and has rapidly become a hot topic of biometric researchers.

The iris of human eye is the annular part between the pupil and sclera. Iris localization is locating the valid part of iris for subsequent feature extraction module. Although state-of-the-art iris features [2-3, 5-6] are very effective for iris recognition, their recognition performance is greatly limited by iris localization. The reasons are as follows:

- The result of iris localization defines the image contents used for feature extraction and matching, which is directly related to the recognition accuracy.
- Real time is a desirable property of iris recognition applications; however, iris localization is a major time-consumer.

Because of the importance of iris localization, a number of algorithms have been developed to achieve accurate and fast iris localization [2-5]. Daugman [2] designed an integrodifferential operator to localize iris; Wildes [3-4] derived the iris boundaries with two approaches, Hough transform and component-goodness-of-fit operator. However, all these methods need to search the iris boundaries over a relatively large parameter space exhaustively, which makes the process time-consuming even a coarse-to-fine strategy is adopted [2]. Besides, they are not very robust and may be trapped in local maximum [3].

To overcome these problems, a novel iterative iris localization algorithm is proposed in this paper, which is inspired by Hooke's law. With a limited number of iterations, the algorithm finds the optimal parameters of the genuine iris boundaries.

2. A brief introduction to the spring

2.1 Hooke's Law

The inspiration of our work stems from the Hooke's law [9]: the restoring force of a spring is directly proportional to its deformation, namely $\vec{F} = -k \cdot \vec{x}$, where $\vec{x}$ measures the displacement of the spring from its equilibrium position, k is the spring constant, the negative sign means the restoring force opposes the deformation.

2.2 An illustration of the mechanics

Figure 1 (a) and (b) illustrate the mechanics of Hooke's springs, where $S_i, i = 1, 2, ...N$, denote N identical massless springs with the equilibrium length R and spring constant k. One end of the springs

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

is attached to a circle whose radius is R, the other end of them joint at point O. At the beginning, all the springs are relaxed, and O is the equilibrium position. Then an appended force is exerted on O to make it move to O' slowly. As a result, each spring produces a restoring force $\vec{f}_i$ to resist the introduced deformation.

$$\vec{f}_i = -k \cdot \vec{x}_i = -k(R - r_i)\vec{e}_i, i = 1, 2, ..., N \qquad (1)$$

where r_i is the current length of S_i and $\vec{e}_i$ is the direction of the ith spring radiating from O'. The composition of these forces $\vec{F}_c = \sum_{i=1}^{N} \vec{f}_i$ will push O' back to its equilibrium position O after the appended force is removed, see Fig.1 (b).

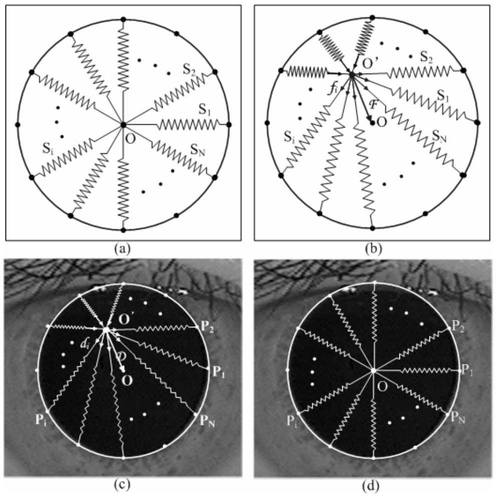

Fig.1 The basic idea of the Pulling&Pushing method.

3. Pulling & Pushing (PP) iris localization

3.1 Basic idea

In this paper, the inner and outer boundaries of the iris are modeled as two non-concentric circles, represented by six independent parameters: the pupil center $O_P(x_P, y_P)$ and pupil radius R_P, the iris center $O_I(x_I, y_I)$ and iris radius R_I.

Take the localization of the inner boundary as an example. Assuming that a rough location of the pupil center $O'(x_P, y_P)$ as well as a set of uniformly sampled edge points on the inner circle ($P_i, i = 1, 2, ..., N$) are available. Join each edge point P_i and the center point O' with an imaginary spring-like line, see Fig.1 (c). Then we get a set of identical 'springs' attached in a circle and jointed at O'. The equilibrium length $\overline{R}$ of these 'springs' is set to $\overline{R} = 1/N \cdot \sum_{i=1}^{N} O'P_i$ and the spring constant is set to $k = 1/N$. Now the same situation appears as in Fig.1 (b): each 'spring' produces a restoring force to resist the introduced deformation, pulling or pushing the current center O' with a displacement $\vec{d}_i$ along each radial direction.

$$\vec{d}_i = \vec{f}_i = -k(\overline{R} - r_i)\vec{e}_i, i = 1, 2, ..., N \qquad (2)$$

The resultant displacement is obtained as $\vec{D} = \sum_{i=1}^{N} \vec{d}_i$, and the center is then moved to $O(x_P, y_P)$. If the edge points and the spring constant of these 'springs' are generated accurately enough, then $O_P = O$ and $R_P = \overline{R}$. However, such is not always the case, so a second pulling&pushing round is performed with the newly estimated center $O(x_P, y_P)$, providing a set of regenerated edge points. This process is repeated until O_P and R_P converge to the genuine position. The flow chart of the pulling & pushing method is depicted in Fig. 2.

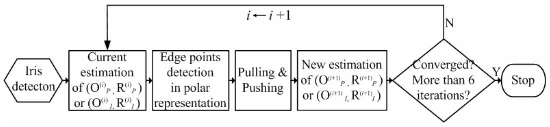
$i \leftarrow i+1$
N
Iris detecton
Current estimation of $(O^{(i)}_P, R^{(i)}_P)$ or $(O^{(i)}_I, R^{(i)}_I)$
Edge points detection in polar representation
Pulling & Pushing
New estimation of $(O^{(i+1)}_P, R^{(i+1)}_P)$ or $(O^{(i+1)}_I, R^{(i+1)}_I)$
Converged? More than 6 iterations?
Y
Stop

Fig.2 The flow chart of the pulling & pushing method.

3.2 Algorithm details

The whole process is divided into four sub-steps: (1) Iris detection, (2) Edge points detection, (3) Pulling & pushing and (4) Convergence criterion. The reminder of this section details each of them.

(1) Iris detection Iris detection provides the initial estimate of the pupil center. Inspired by the work of Viola et al. [9], a cascade classifier based on AdaBoost is developed to detect the existence of the iris in the input images. After iris detection, a rough estimate of the pupil's center $O_P^{(0)}(x_P^{(0)}, y_P^{(0)})$ is extracted in iris images while non-iris images are excluded. An example of $O_P^{(0)}$ and $R_P^{(0)}$ is shown in Fig. 3 (a), where $R_P^{(0)}$ is a preset constant. Here and throughout the paper, the number in the superscript represents the index of iteration.

(2) Edge points detection Edge points detection is accomplished in the polar coordinate. With the intuitive insight that a circle centered on the origin in Cartesian coordinate will be a vertical line in polar coordinate, the complicated circular edge detection problem is thus transformed to a simpler line edge detection one. $O_P^{(0)}(x_P^{(0)}, y_P^{(0)})$ is chosen as the origin of the transform.

The detection can be done by various techniques such as Sobel operator or Canny edge detector [8]. Only one edge point is reserved in each radial direction, denoted by $P_i^{(1)}(\alpha_i^{(1)}, r_i^{(1)}), i = 1, 2, ..., N$ in polar representation, where $\alpha_i^{(1)}$ is the *i*th angle, $r_i^{(1)}$ is the distance from $O_P^{(0)}(x_P^{(0)}, y_P^{(0)})$. These edge points are labeled with light spots in Fig.3 (b).

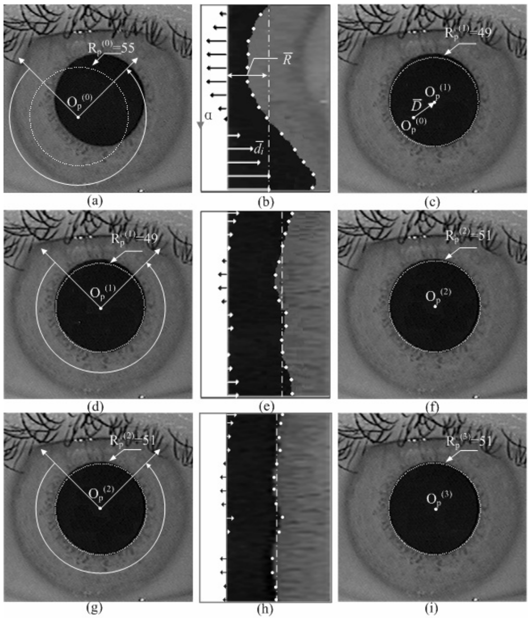

Fig.3 An illustration of the Pulling&Pushing procedure. Each row represents one iteration. Column 1 shows the current estimate of the pupil center and its radius; column 2 shows the detected edge points and the displacements the center should move along each radial direction (in polar representation). The new estimate of the pupil center and its radius is shown in column 3.

(3) Pulling & Pushing Once the center estimate $O_P^{(0)}(x_P^{(0)}, y_P^{(0)})$ and N edge points $P_i^{(1)}(\alpha_i^{(1)}, r_i^{(1)}), i = 1, 2, ..., N$ are available, the equilibrium length $\overline{R}_P^{(1)}$ of these 'springs' is then obtained by

$$\overline{R}_P^{(1)} = 1/N \sum_{i=1}^{N} r_i \tag{3}$$

which is taken as an estimate of the pupil radius.

Besides, each 'spring' will pull or push the current center $O_P^{(0)}$ with a displacement, $\vec{d}_i$ as illustrated in Fig.1 (c) in Cartesian representation and Fig.3 (b) in polar representation. The resultant displacement of the center point is obtained by

$$\vec{D} = \sum_{i=1}^{N} \vec{d}_i \tag{4}$$

forcing the pupil center to move to,

$$O_P^{(1)}(x_P^{(1)}, y_P^{(1)}) = O_P^{(0)} + \vec{D} \tag{5}$$

as illustrated by Fig. 3 (c), and $O_P^{(1)}$ is taken as a new estimate of the pupil center.

In all, a new estimate of the R_P and $O_P(x_P, y_P)$ is obtained via Eqs. (3) and (5).

(4) Convergence criterion Generally speaking, $O_P^{(1)}$ and $\overline{R}^{(1)}$ are still not the genuine position of the pupil center and the radius, so the above steps (2) and (3) must be repeated with the new estimate of the pupil center, until either of the two criteria is satisfied: (a) $O_P^{(i)}$ and $\overline{R}_P^{(i)}$ converge; and (b) the number of iteration is above a fixed threshold $I_{\max}$.

For (a), a convergence criterion function is developed,

$$\mathrm{C(i)} = \left|\mathrm{O_P^{(i)} - O_P^{(i-1)}}\right| + \left|\overline{\mathrm{R}}^{(i)} - \overline{\mathrm{R}}^{(i-1)}\right| = |\vec{D}| + \left|\overline{\mathrm{R}}^{(i)} - \overline{\mathrm{R}}^{(i-1)}\right| \tag{6}$$

The procedure converges if C(i) is less than a threshold $C_{\max}$ (set to one pixel here). Criterion (b) avoids PP procedure being trapped in an infinite cycle.

The PP procedure is illustrated in Fig. 3 step by step.

4. Experimental results

CASIA Iris Database [10] is adopted to evaluate the performance of the pulling & pushing algorithm. This database includes 1200 640*480 iris images from 60 different eyes, with the diameter of the iris ranging from 180 to 260 pixels. The experiments are done in Matlab on a PC with 2.4G processor and 512M DRAM.

4.1 Accuracy and speed

On the above database, the Pulling & Pushing method is proven to be very accurate. Daugman's integrodifferential method [2] is also implemented on the same database for a comparison.

Table 1. Performance of the PP method

	Iris Detection	Integrodifferential	Pulling & Pushing
Accuracy (%)	100	97.3	99.6
Speed (ms)	84	675	156

The experimental results are given in Table 1, from which we can see that the PP method results in more accurate localization and more rapid execution. Note that the experiments are done under the condition that the same initial estimate of the pupil center is provided for both methods. The localization results of three examples are shown in Fig.4.

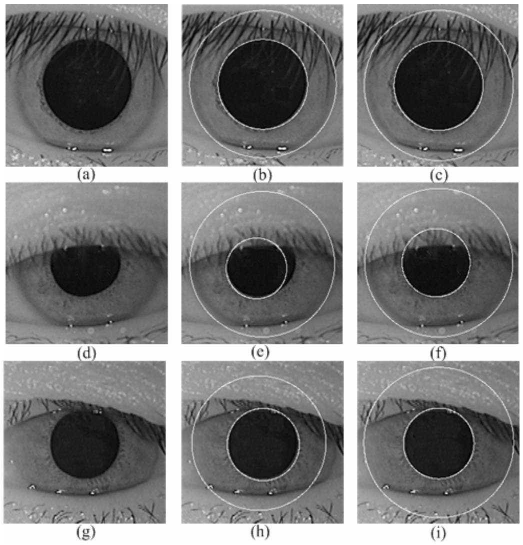

Fig.4 Localization examples. Column 1 shows the original images; Column 2 and Column 3 give the localization result of the Daugman's method and the PP method respectively. Both methods achieve high accuracy in Row 1. However, Daugman's method shows imperfect localization when facing deformed pupil, low contrast iris-sclera boundary and eyelid occlusion (Row 2 and 3), while the PP method appears more robust to such challenges.

4.2 Remarks

Although encouraging performance has been obtained, several points should be noticed.

1. The initial estimate of the pupil's center. This is crucial to the proposed method. The least essential requirement is that the initial point must be located in the pupil region.

2. The setting of the 'spring' constant k . A small k will slow the convergence speed, while a large k may even induce divergence.

3. Edge points detection. The more accurate the edge points, the faster and more accurate the localization.

5. Conclusions

In this paper, a novel iris localization algorithm has been presented. Inspired by the mechanics of Hooke's law, this algorithm finds the optimal path towards the genuine iris boundaries via pulling and pushing. Experimental results have shown that the Pulling & Pushing method results in high accuracy and rapid implementations, and satisfies the requirements of real application.

Our future work will focus on occlusion detection based on the result of Pulling & Pushing method.

Acknowledgement

This work is partly supported by the National Basic Research Program (Grant No. 2004CB318110), the National Natural Science Foundation of China (Grant No. 60335010 and No. 60572058), MST of PRC (Grant No.2004DFA-06900) and the Chinese Academy of Sciences.

Reference

[1] A. K. Jain, A. Ross and S. Prabhaker, "An introduction to Biometric Recognition", *IEEE Trans.* CSVT, 2005, Vol. 14, No. 1, pp. 4-20.

[2] J. Daugman, "Statistical Richness of Visual Phase Information: Update on Recognition Persons by Iris Patterns", *IJCV*, 2001*,* Vol. 45, No. 1, pp.25-38.

[3] R. Wildes, "Iris Recognition: An Emerging Biometric Technology", *Proc. of the IEEE*, 1997, Vol.85, pp.1348-1363.

[4] T. A. Camus and R. Wildes, "Reliable and Fast Eye Finding in Close-up Images", *Proc. ICPR*, 2002, pp.389-394.

[5] Li Ma, Tieniu Tan, Yunhong W., Dexin Zhang, "Efficient Iris Recognition By Characterizing Key Local Variations", *IEEE Trans. IP*, 2004, Vol. 13, NO.6, pp.739–750.

[6] Zhenan Sun, Tieniu Tan, Yunhong Wang, "Robust Encoding of Local Ordinal Measures: A General Framework of Iris Recognition", *ECCV* 2004.

[7] Paul Viola and Michael J. Jones, "Robust real-time object detection", *Proc. of IWSCT* 2001.

[8] Rafael C. Gonzales , Paul Wintz, "*Digital Image Processing"* (2nd ed.), 1987, Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

[9] Young, Hugh D., and Roger A. Freedman. *University Physics with Modern Physics*. 11th ed. San Francisco, CA: Addison Wesley, July 25, 2003.

[10] CASIA Iris Image Database, http://www.sinobiometrics.com