MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

A System for Automatic Iris Capturing

Guodong Guo, Michael Jones and Paul Beardsley

TR2005-044 December 2005

Abstract

Biometrics is increasingly important in security applications. Iris recognition provides the greatest accuracy among known biometrics. The accuracy of iris recognition is, for example, much greater than face recognition and fingerprint recognition. However, it is not trivial to capture iris images in practice, and usually the users need to adjust their eye positions for iris image acquisition (e.g., the classical Daugmanś and Wildes systems). This paper describes a new system to capture iris images automatically without user interaction. It works at a distance of over l meter to the users. Experimental results demonstrate the performance of the system.

A System for Automatic Iris Capturing

Guodong Guo
University of Wisconsin-Madison
gdguo@cs.wisc.edu

Michael J. Jones
Mitsubishi Electric Research Labs
mjones@merl.com

Paul Beardsley
Mitsubishi Electric Research Labs
beardsley@merl.com

Abstract

Biometrics is increasingly important in security applications. Iris recognition provides the greatest accuracy among known biometrics. The accuracy of iris recognition is, for example, much greater than face recognition and fingerprint recognition. However, it is not trivial to capture iris images in practice, and usually the users need to adjust their eye positions for iris image acquisition (e.g., the classical Daugman's and Wildes' systems). This paper describes a new system to capture iris images automatically without user interaction. It works at a distance of over 1 meter to the users. Experimental results demonstrate the performance of the system.

1 Introduction

A wide variety of systems require reliable personal identification or verification. Biometric technology overcomes many of the disadvantages of conventional identification and verification techniques such as keys, ID cards and passwords. Biometrics refers to the automatic recognition of individuals based on their physiological and/or behavioral characteristics [5]. There are many features to use as biometric cues, such as face, fingerprint, hand geometry, handwriting, iris, retinal, vein, and voice. Among all these features, iris recognition gives the highest accuracy [7]. The complex iris texture carries very distinctive information. Even the irises of identical twins are different [2] [5].

However, it is difficult to capture iris images in practice. Classical iris recognition systems, e.g., Daugman's and Wildes', need the users to adjust their eye positions in order to capture their irises [11]. Furthermore, the classical systems require users to be close to the capturing apparatus [1] [12] [6]. Hence, to design an iris capturing system which works without user interaction is of great importance in practice. This is the motivation of our work.

In this paper, we present a system for capturing iris images automatically without user interaction from a distance of 1.2 meters to 2.1 meters. The paper is organized as follows. In Section 2, the system hardware is described. Then the system software is described in Section 3. The experimental setting and results are provided in Section 4. Finally, some discussions and conclusion are given.

2 System Hardware

The system has two cameras, one is a video camera with wide field of view (WFOV) and the other is a high resolution digital still camera with narrow field of view (NFOV). The video camera captures any change in the view, e.g., a face appears, while the high resolution digital still camera is used to acquire the iris images. In order to move the two-camera system so that the eye regions are aligned to the NFOV still camera, a pan-tilt-unit (PTU) is used.

2.1 Camera Selection

The camera with WFOV captures any change in the viewing field, e.g., whether there is a face or not, and more importantly provides temporal information, e.g., whether and when an eye region is ready for iris capturing by another camera. For these purposes, a video camera must be used for the WFOV camera. On the other hand, there is no need to use temporal or motion information in iris recognition [1] [2] [11] [6]. The traditional approaches first capture a short video sequence of iris and then choose one frame with the best quality [2] [6] [8]. Since temporal information is not necessary and since current video cameras do not capture very high resolution images, we have chosen a high resolution digital still camera for the NFOV camera.

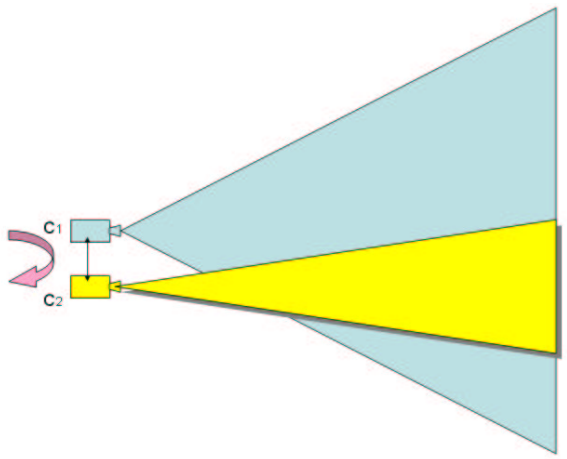

Figure 1: The two camera system setup. C_1 is the video camera with WFOV, while C_2 is the high resolution digital still camera with NFOV. Two cameras are fixed and can move together by a PTU.

2.2 Configuration

The system setup is illustrated in Figure 1 with the following properties:

- The two cameras are close together with approximately parallel optical axes. This setting guarantees a face appears as a frontal view in both cameras.
- The two cameras are rigidly attached, and the spatial relation between two camera image planes is precalibrated.
- The two cameras are mounted on a pan-tilt-unit (PTU). In order to move the cameras so that the eye regions are viewed by the high resolution still camera, a PTU is used to support the cameras. Pan and tilt commands are issued when the eye regions are out of the field of view of the high resolution still camera.

3 System Software

In the previous section, we described the hardware setup and the basic operation for the iris capturing system. Now, we will explain the algorithms needed to run the system. The system block diagram is shown in Figure 2, and each block will be described specifically.

The basic operation of the system is to continuously look for a face in the WFOV video image until one is found. Then some facial feature points are located and a tightly cropped bounding box around the eyes is computed. This eye region is mapped into the image plane of the NFOV still camera. If the eye region is well centered in the NFOV still camera then a photo of the eyes is captured. If the eye region is not well centered then the PTU is used to pan and tilt both cameras so that the eye region becomes closer to the center of the still camera's image plane. The detection of faces and features and subsequent panning and tilting of the cameras is iterated until the eye region is well centered in the still camera's image plane.

We describe the components of this software system in more detail below.

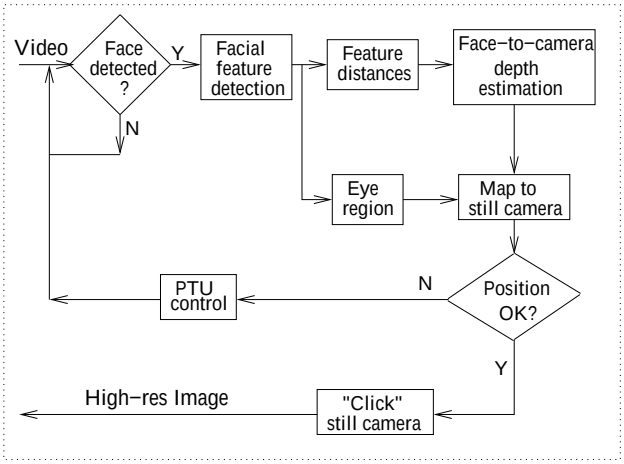

Figure 2: The system block diagram. The input is the video images and the output is the captured high resolution iris image. See text for details.

3.1 Face Detection

To capture the iris, the system first has to know whether there is a person in the scene. The video camera is used for this purpose. For each video image frame, a face detector is called to find faces. We use the face detector described in [10] which is based on a cascade of classifiers that use simple Haar-like features originally selected using the AdaBoost learning algorithm [3]. The advantage of this face detector is its speed and robustness.

3.2 Facial Feature Detection

After a face is detected in the video image frame, some facial features are also detected within the face box. We use the same Viola-Jones style detectors for detecting 9 facial feature points: left and right outside eye corners, left and right eye centers, left and right nose corners, nose tip, center of upper lip, and the bridge of the nose. See Figure 3 or 6 for the nine detected features (each displayed with a white square).

The training examples for each of the facial feature detectors are simply rectangular regions around each feature where each feature location has been precisely specified by hand. The only difference between the feature detectors and the face detector is the examples each was trained on.

The detected facial features have two purposes in the iris capturing system: one is to estimate the depth of the face to the video camera, and the other is to compute the bounding box of the eye region.

3.3 Depth of Face to Camera

After a face is detected in the video frame, the system needs to know the depth of the face to the camera. This is so that the eye region can be mapped into the image plane of the still camera to decide whether to snap a photo or control the pan-tilt unit. Traditionally, a laser range finder or stereo techniques (using two or more cameras) can be used to obtain the depth of an object to the camera. This adds further cost and complexity to the system.

Here we present a technique that only uses the video camera to compute the depth. We use the facial features directly to estimate the face depth to the camera. This technique is based on the geometric optics of a pin-hole camera model: the image of an object is bigger if the object is closer to the camera, and vice versa. Using this property, a mapping from facial feature distances to depth values is learned.

3.3.1 Independent Linear Regression

Assume we collect a data set of n faces at four different depths from the video camera. For each face we compute N facial feature distance measures. Let $d^i_{j,k}$, $1 \le j \le N$, $1 \le i \le n$ be the Euclidean distance between the j^{th} pair of feature points for face i at depth index k. D_k is the depth for discrete depth index k, $1 \le k \le 4$. We will use linear regression to map each feature distance, $d^i_{j,k}$ to the depth from the video camera :

$$a_j \cdot d^i_{j,k} + b_j = D_k.$$

To compute a_j and b_j for each distance feature j we need to solve a set of linear regressions

$$A_j \cdot X_j = 0 \tag{1}$$

with

$$A_j = \begin{bmatrix} d^1_{j,1} & 1 & -D_1 \\ \vdots & \vdots & \vdots \\ d^n_{j,1} & 1 & -D_1 \\ d^1_{j,2} & 1 & -D_2 \\ \vdots & \vdots & \vdots \\ d^n_{j,2} & 1 & -D_2 \\ d^1_{j,3} & 1 & -D_3 \\ \vdots & \vdots & \vdots \\ d^n_{j,3} & 1 & -D_3 \\ d^1_{j,4} & 1 & -D_4 \\ \vdots & \vdots & \vdots \\ d^n_{j,4} & 1 & -D_4 \end{bmatrix} \tag{2}$$

and

$$X_j = \begin{bmatrix} a_j \\ b_j \\ 1 \end{bmatrix} \tag{3}$$

Hence, there is a different linear mapping from feature distance to camera depth for each different pair of features. It is straightforward to solve Equation. (1) using the singular value decomposition (SVD) technique.

Since each feature is processed independently, we call this method independent linear regression (ILR). To get a single depth estimate, all of the depth estimates are averaged. Thus, from a set of feature distances, $\{d_l\}$ the corresponding linear mappings for each feature distance are used to get a set of estimated depths, $\{\Delta_l\}$:

$$a_l \cdot d_l + b_l = \Delta_l, \;\; l \in \{1, \cdots, L\} \tag{4}$$

$$\bar{\Delta} = \frac{1}{L} \sum_{l=1}^{L} \Delta_l \tag{5}$$

where L is the real number of feature distance measures for a test face with $L \le N$. When some features are not detected, $L < N$.

This results in a more robust estimate than using only the distance for a single pair of features. It also has the advantage of easily handling missing feature points. When a feature is not detected, the linear mapping for that distance is simply not used, and the depth estimates from all the other distance measures are averaged to yield a robust depth measure.

Using the ILR method, the procedure of depth estimation in both the learning and running phases are given below.

3.3.2 Learning Phase

- Divide facial features into groups. In our case, nine facial feature points are detected in each face image. Because the image distance measure is sensitive to close

feature points, the nine points are partitioned into 4 groups in order to get a robust estimate. See Figure 3 for an illustration.

- Compute the pairwise Euclidean distances from a point in one group to all points in other groups.
- Concatenate distance measures into a feature vector. In our case, 28 distance measures are computed given this 4-group-division of nine facial features. The resulting feature vector is of dimension 28.
- Repeat above processes for various faces captured by the video camera at various depths to the cameras.
- Compute regression coefficients a_j and b_j using the ILR method.

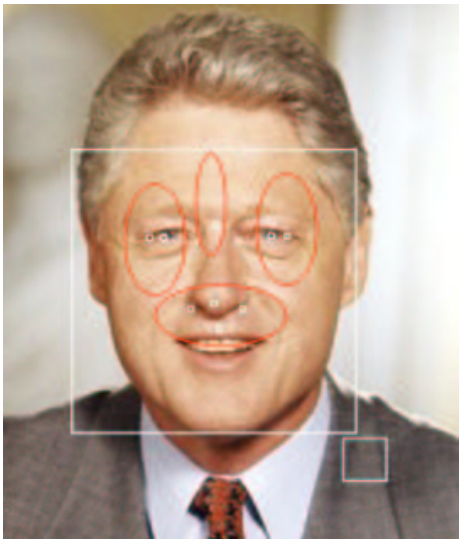

Figure 3: Facial features (9 white squares) detected within the face box. They are divided into 4 groups for pairwise feature distance measure. Note a small box near the collar is detected as a face, but no features are detected inside, which shows that feature detection is also useful for correcting false positive face detections.

3.3.3 Running Phase

For a new face in running stage, the system first detects face and facial features. Then the pairwise distance measures are computed with the same 4 group division as in the learning stage. The regression coefficients a_j and b_j are used to estimate the depth of face to the camera using equations (4) and (5). In practice, it is possible to obtain fewer than 28 distance measures (e.g., missing data problem), but the ILR algorithm can easily deal with this.

3.4 Calibration

The goal of calibration is to enable the eye positions detected in the video image to be mapped to estimated eye positions in the still image. One way to achieve this would be to do a full Euclidean stereo calibration of the video camera and still camera. Given full calibration and an estimate of the depth of the face from the video camera (see Section 3.3), it is straightforward to find the face position in the still camera. But the still camera is autofocus, and a full Euclidean calibration would be arduous[13]. We adopt a simpler partial calibration that is completely adequate for our goal.

First note that if the face is at a known depth d from the cameras, then the calibration is simple. A homography is computed for a fronto-parallel plane at depth d from the cameras. A plane is an approximate model for the face, so the homography describes approximately the mapping of features on the face between the two cameras. Assuming that the face has a known depth is acceptable for some types of situation, for example when a user is instructed to stand at a line on the floor at an access point (this constrained situation is still more flexible than systems that require the user's head to be in a specific position).

Now consider the case when the face lies within some range of depths. The range is quantized, and a separate homography is computed for a fronto-parallel plane at each depth $d_1, d_2 ... d_n$. At run-time, the distance d to the face is estimated, and the homography associated with the distance d_i that is closest to d could be used to provide the mapping of face features between the cameras. Alternatively, we can interpolate the calibrated homographies to find a mapping for facial features at depth d, as described in Section 3.5.

For computing the homography, we use a calibration plane with the pattern shown in Figure 4 The video camera captures the full pattern, and feature points are found automatically for the large squares. The still camera has a narrower field of view and captures just the central three-by-three grid of small squares, and features points are found automatically for these squares. Knowing these image feature points and the Euclidean coordinates of the full pattern, it is straightforward to find homography H_{VP} between the video image and the pattern, and homography H_{SP} between the still image and the pattern, and hence homography $H_{VS} = H_{SP}^{-1} H_{VP}$ between the video image and the still image [4]. H_{VS} is a 3×3 matrix that describes the mapping of a homogeneous feature point x_v in the video image to a point x_s in the still image by

$$x_s = H_{VS} x_v \tag{6}$$

As described above, the process is repeated for a set of depths of the calibration pattern from the cameras, to give a set of homographies $H_{VS1}, H_{VS2}, H_{VSn}$.

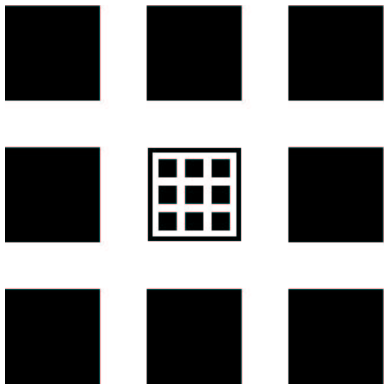

Figure 4: Calibration pattern used for computing the homography between two image planes. The wide-FOV video camera captures the entire pattern, while the narrow-FOV still camera captures the central three-by-three grid of small squares.

3.5 Cross Ratio, Projective Invariance

Assume at run-time that the face is at depth d from the cameras. This section describes a simple technique to interpolate between the homographies H_{VSi} at depths d_i to determine a mapping between the video and still cameras for features at depth d.

The cross ratio of four numbers is invariant under a general homography [9]. For a line AD shown in Figure 5, the cross ratio is defined as $cr = \frac{AB}{BD} / \frac{AC}{CD}$, which equals $\frac{A'B'}{B'D'} / \frac{A'C'}{C'D'}$.

How do we use the cross ratio in our two camera system? In Figure 5, C_1 and C_2 are two camera centers. Let I_1 be the video image plane, and I_2 be the high resolution still camera image plane. For any image pixel in I_1, there is a viewing line, e.g., C_1A. If the homography from I_1 to I_2 at depth A is known, we can map the real 3D point at A to A' in image plane I_2. Similarly, the 3D points at C and D can be mapped to C' and D', respectively, assuming the homographies at depths C and D are known. Suppose the homography at B is unknown. Using the technique in Section 3.3, the depth of B can be estimated. Thus the cross ratio cr of A, B, C and D in line AD can be computed. Then the cross ratio cr is used for line $A'D'$ based on the invariant property.

Actually, the coordinates of B', (x_b, y_b), in I_2 can be obtained by equations,

$$x_b = \frac{cr \cdot x_c \cdot x_d + (1 - cr) \cdot x_a \cdot x_d - x_a \cdot x_c}{x_d - (1 - cr) \cdot x_c - cr \cdot x_a} \quad (7)$$

$$y_b = \frac{cr \cdot y_c \cdot y_d + (1 - cr) \cdot y_a \cdot y_d - y_a \cdot y_c}{y_d - (1 - cr) \cdot y_c - cr \cdot y_a} \quad (8)$$

where (x_a, y_a), (x_c, y_c), (x_d, y_d) are the coordinates of A', C', and D' in image plane I_2, and they are computed using the pre-calibrated homographies at known depths A, C, and D. Although we actually have 4 precomputed homographies at known depths, we only use 3 of these with the cross ratio.

In this way, any point in image plane I_1 can be mapped to I_2 at any depth to the cameras.

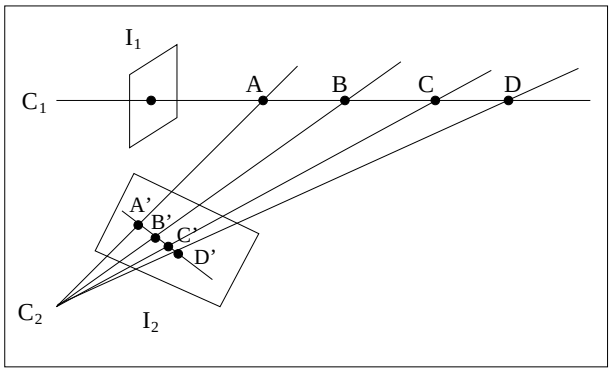

Figure 5: Cross ratio computation in the two camera system setup. See text for details.

3.6 Eye Region via Facial Feature Points

To capture the high resolution iris images, the system first needs to know where the eye region is. The facial features can be used to determine the eye region. As shown in Figure 6, a simple strategy is to use 2 eye corners to determine the eye region. Assume the distance between two eye corners is d_1, let $W = 1.25 \times d_1$ and $H = 0.5 \times W$, where W and H are the width and height of the eye region, then we have

$$\begin{aligned} X_l &= X_1 - \frac{5}{32} \times d_1 \\ X_r &= X_2 + \frac{5}{32} \times d_1 \\ Y_l &= Y_1 - \frac{5}{16} \times d_1 \\ Y_r &= Y_2 + \frac{5}{16} \times d_1 \end{aligned}$$

where (X_1, Y_1) and (X_2, Y_2) are the image coordinates of the left and right eye corners, and (X_l, Y_l) and (X_r, Y_r) are the coordinates of the upper-left and bottom-right corners of the eye region rectangle.

The eye region in the video image plane I_1 can be mapped to the high resolution image plane I_2 using the technique presented in Section 3.3 to 3.5.

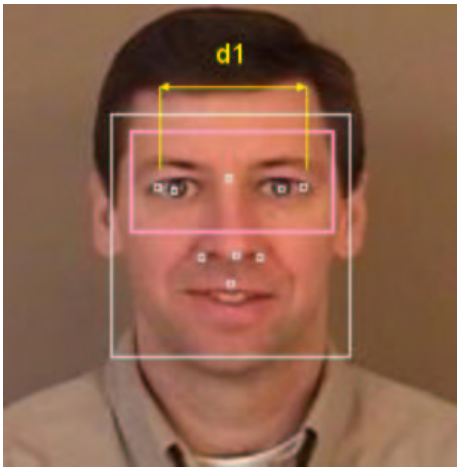

Figure 6: Facial features detected determine the eye region in the video image. The outer box is the face detection result, while the inner rectangle is the computed eye region in the face image. d_1 is the Euclidean distance between two eye corners.

3.7 Pan-Tilt Unit Control and Eye Capture

The eye region is normally contained in the video images, but usually not in the view of the high resolution still camera, because of the variations of the users' heights and standing positions to the left or right. The high resolution still camera ought to move so as to view each person's eyes in any case.

The eye region in the video image is mapped to the high resolution image. If the region is contained in the high resolution image plane and the center of the mapped eye region is close to the image center, the high resolution still camera immediately captures an image that includes the eyes. Otherwise, the system pans and tilts the cameras iteratively for each video frame until the eye region is approximately centered in the high resolution still camera.

4 Experiments

For the video camera, we use a Sony DCR-PC105. For the high resolution digital still camera, we use a Canon Digital Rebel which has a resolution of 3072×2048 (6 megapixels). To capture the iris images at a distance, a telephoto lens EF70-200mm is added to the camera. The minimum shooting distance for the telephoto lens is 1.2 meters, thus the iris images are captured at least 1.2 meters away.

To estimate the linear mapping from feature distances to camera depth described in section 3.3.2, 10 persons were asked to stand at four different distances: 1.2, 1.5, 1.8, and 2.1 meters from the cameras. Then the video camera captured image frames of faces. We captured a total of 40 face images - 4 images per person. Face detection and facial feature detection are executed for these face images. The ILR algorithm was then used to learn the linear mappings for depth.

To compute the homographies at four different depths from the cameras, we fixed the calibration pattern at the same four depths: 1.2, 1.5, 1.8, and 2.1 meters. The method described in Section 3.4 was then used to compute the homographies.

To determine the eye region for each face, 5 images are randomly chosen from the 40 images used for depth learning, and the relation of the eye region size and the distance between two eye corners is examined in the 5 images. We found the approximation shown in Section 3.6 works well in practice.

We tested the system on a number of people and found that the person's eye region is typically captured within 2 to 5 seconds. It is possible to improve the control strategy of the pan-tilt unit to make it much faster.

The system works well in automatically capturing high resolution images of both irises. Two examples are shown in Figs. 7 and 8. In both figures, the full high resolution image automatically captured is shown, along with a zoomed image of the right eye which better shows the level of detail captured in the high resolution image. The texture in the iris regions is clear visually. All irises have at least 200 pixels in diameter, which is enough resolution for iris recognition [2].

Furthermore, we capture both eyes with the benefit of the high resolution still camera. Two eyes can provide contextual information for each other in recognition. For example, it is not necessary to try to match a large number of possible in-plane rotations as many current approaches do [6]. Since we capture both irises and can estimate the center of each iris, we can compute the in-plane rotation and derotate the image to zero degrees. Note that it is impossible to capture both eyes using traditional video cameras because of the lack of resolution.

5 Discussion

Almost all previous systems use video cameras to capture the iris images. In our system, the high resolution still camera is used instead. The advantages are 1) both eyes can be captured simultaneously without sacrificing resolution. Two eyes may provide contextual information for each other in iris recognition; 2) only one image is captured in focus. Previous systems usually use a video camera to capture a short video sequence and then choose one frame for recognition [2] [6].

A comparison between our iris capturing system (referred to as the MERL system in the table) and some rep-

Table 1: A comparison of our system (MERL system) with previous iris capturing systems. They are compared by capturing distance from the eye to camera, the camera type for iris capturing, number of cameras used in the whole system, requiring user interaction or not, and the number of eyes can be captured simultaneously.

Systems	Cap. Dis.	Iris Cam.	#Cams.	User Inter.	#eyes
Daugman's [1]	15-46 cm	Video	1	Yes	1
Wildes' [12]	20 cm	Video	1	Yes	1
Sensar [8]	38-76 cm	Video	3	No	1
Tan's [6]	4 cm	Video	1	Yes	1
Panasonic BM-ET 300	30-40 cm	Video	1	Yes	1
Panasonic BM-ET 500	30-60 cm	Video	1	No	1
MERL System	120-180 cm	High-res Dig.	2	No	2

resentative systems is given in Table 1. From the table, one can see that all previous systems need the user to be close to the camera, but our system can operate from $120cm$ to $180cm$. In Sensar's system [8], three cameras are used, two for stereo computation, and the third for iris capturing. They need special hardware for stereo computation, which is not needed in our system. Furthermore, only our system can capture both eyes which provide contextual information for each eye in iris recognition. The performance of the Panasonic iris systems are obtained from their product manuals.

In the current system, we have not used near-infrared illumination which is useful for capturing black eyes, but that is not difficult to add to the system. Our main goal here is to capture the eyes automatically.

6 Conclusion

We have developed a system for automatic iris capture without user interaction. The system works at a distance of over one meter to the user and is very robust. Only 2D image data is used without involving complex 3D computation. Our work takes a step towards more practical use of the iris as a biometric.

7 Acknowledgements

Thanks to John Barnwell for helping us build a rig to fix the cameras to the pan-tilt unit.

References

[1] J. G. Daugman. High confidence visual recognition of persons by a test of statistical indenpendence. In *IEEE Patt. Anal. Mach. Intell.*, volume 15, pages 1148–1161, 1993.

[2] John Daugman. How iris recognition works. *IEEE Trans. on Circuits and Systems for Video Technology*, 14:21–30, 2004.

[3] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.

[4] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[5] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14:4–20, 2004.

[6] L. Ma, T. Tan, Y. Wang, and D. Zhang. Personal identification based on iris texture analysis. In *IEEE Patt. Anal. Mach. Intell.*, volume 25, pages 1519–1533, 2003.

[7] T. Mansfield, G. Kelly, D. Chandler, and J. Kane. Biometric product testing final report. *UK Biometric Work Group Report*, 2001.

[8] M. Negin, T. Chmielewski, M. Salganicoff, U. von Seelen, P. Venetainer, and G. Zhang. An iris biometric system for public and personal use. In *IEEE Computer*, volume 33, pages 70–75, 2000.

[9] C. E. Springer. *Geometry and Analysis of Projective Spaces*. W. H. Freeman and Company, 1964.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[11] R. Wildes. Iris recognition: An emerging biometric technology. *Proc. IEEE*, 85:1348–1363, 1997.

[12] R. P. Wildes, J. C. Asmuth, G. L. Green, S. C. Hsu, R. J. Kolczynski, J. R. Matey, and S. E. Mcbride. A system for automated iris recognition. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 121–128, 1994.

[13] Reg G. Willson and Steven A. Shafer. A perspective projection camera model for zoom lenses. In *Proceedings Second Conference on Optical 3-D Measurement Techniques*, October 1993.

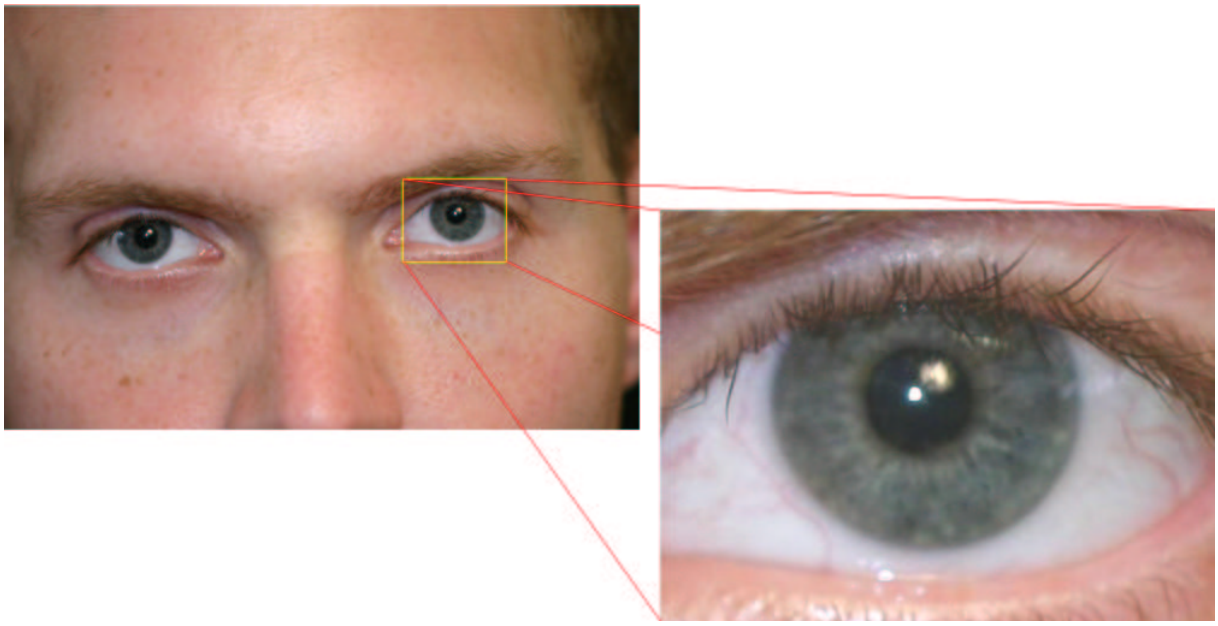

Figure 7: An example of the captured high resolution eye regions. The image is of size 3072×2048. The right eye is shown for visual inspection. For a better view, please look at the image in the electronic file instead of the printed one.

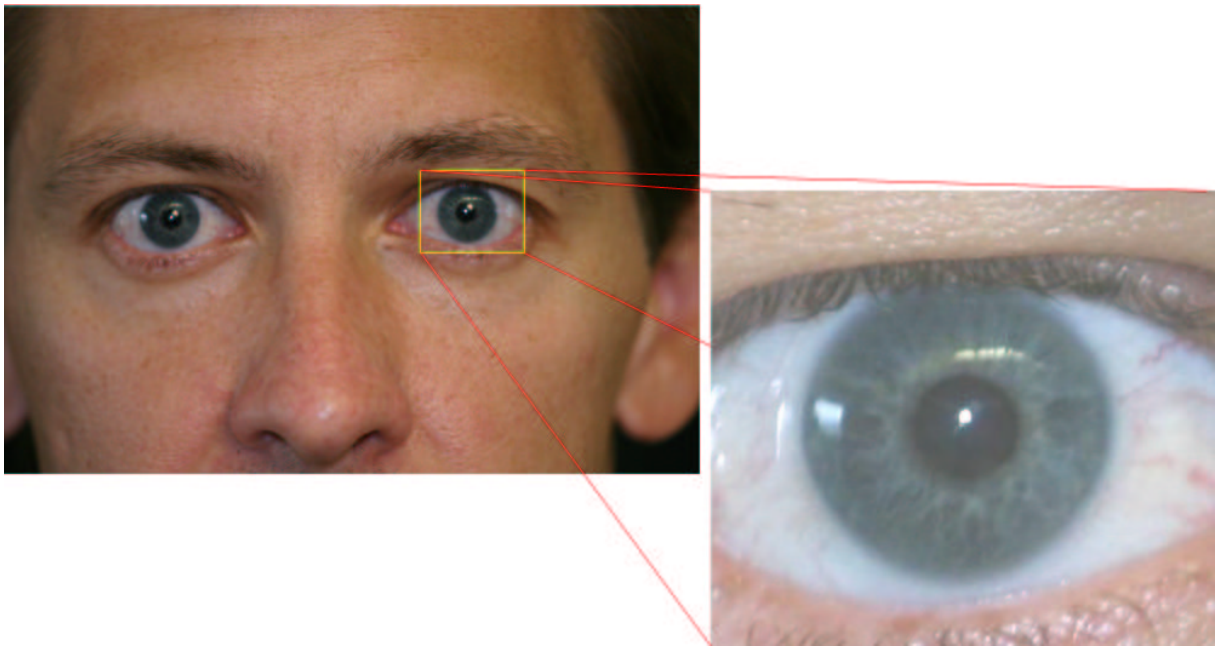

Figure 8: Another example of the captured high resolution eye regions.