Preprocessing Methods for SVD-based Iris Recognition

Petr Gajdoš, Pavel Moravec, Václav Snášel
Department of Computer Science, FEECS
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
{petr.gajdos, pavel.moravec, vaclav.snasel}@vsb.cz

Abstract

In this paper, we compare the performance of singular value decomposition for the recognition of human irises after image preprocessing by quaternion transformation and colour levelling based on contour detection. The results are also compared with our previous approach based on classic iris unfolding.

Keywords: SVD, information retrieval, iris recognition, quaternion, contours

1 Introduction

Methods of human identification using biometric features like fingerprint, hand geometry, face, voice and iris are widely studied.

A human eye iris has its unique structure given by pigmentation spots, furrows and other tiny features which are stable throughout life. It is possible to scan an iris without physical contact in spite of wearing contact lenses or eyeglasses. The iris is hard to forge which makes the iris a suitable object for the identification of people. Iris recognition seems to be more reliable than other biometric techniques like face recognition [5]. Iris biometrics systems for both private and public use have been designed and deployed commercially by NCR, Oki, IriScan, BT, US Sandia Labs, and others.

In this paper, we use the Petland's approach to image retrieval: image vectors of complete images of the size width $\times$ height of the image [17] build the feature vectors.

Singular value decomposition (*SVD*) was already successfully used for automatic feature extraction. In case of face collection, the *base vectors* can be interpreted as images, describing some common characteristics of several faces. These base vectors are often called eigenfaces. For a detailed description of eigenfaces, see [17]. Based on the same convention, we will call the base vectors obtained from irises eigenirises.

In this paper we will concentrate on preprocessing the irises before the calculation of SVD which should help us to obtain a better iris identification and compare it with our previous results [11, 12].

The rest of this paper is organized as follows. The second section explains unfolding, the third describes used preprocessing methods. In the fourth section mentions dimension reduction by singular value decomposition, in next section we briefly describe qualitative measures used for evaluation of our tests. The rest of the paper contains a description of the collection, results of tests and conclusions with ideas for future research.

2 IRIS Unfolding

Following section describes a process of IRIS unfolding. The whole process is divided into several parts which can be differently implemented with respect to selected hardware and software. OpenGL API was used to get better results in our case. It allows us to process whole IRIS collection in a shorter time with respect to variable precision of image unfolding. Next, this solution is prepared for further implementation on GPUs (Graphic Processor Units) which becomes standard in the area of image processing.

- First, inner and outer boundaries of the iris must be detected. This can be done manually but some of image processing methods (i.e. histograms) can be used as well to process this step automatically.

- Since the boundaries have been detected, the iris ring is divided into several segments (see the Figure 1). Lower precision of resulting image is caused by small number of segments. Therefore, the number of segments should be set with respect to input image resolution. The upper limit is represented by the number of points of the outer iris boundary. Should the number of segments be greater then the upper limit, then redundant points appear in resulting image. The illustration example has eight segments (see the Figure 2).

- The corners of all segments represents texture coordinates. The segments are mapped on rectangular areas by the usage of OpenGL (see the Figure 3). The figure also illustrates the lost precision at the top boundary which is given by texture mapping and image dilatation. This is prevented by higher number of segments as it was mentioned in the previous point.

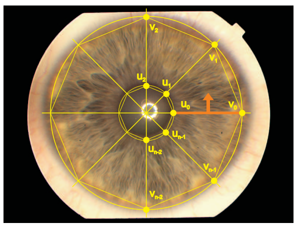

Figure 1. The first phase of IRIS unfolding

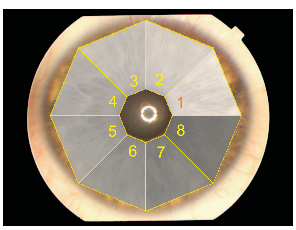

Figure 2. Iris segments

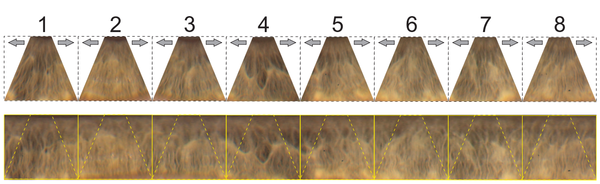

Figure 3. Segment mapping

3 Preprocessing methods

3.1 Quaternions

According to [4], quaternions were invented by Sir William Rowan Hamilton in 1843. His aim was to generalize complex numbers to three dimensions, i. e. numbers in the form $a + \mathbf{i}b + \mathbf{j}c$, where $a, b, c \in R$ and $\mathbf{i}^2 = \mathbf{j}^2 = -1$. Hamilton has never succeeded in making this generalization, and it was later proven that the set of three-dimensional numbers is not closed under multiplication. One of Hamilton's motivations was to find a description of rotation in space corresponding to the complex numbers, where a multiplication corresponds to a rotation and a scaling in the plane. Later Hamilton realized that four numbers are needed to describe a rotation followed by a scaling. One number describes the size of the scaling, second the number of degrees to be rotated, and the last two numbers give the plane, in which the vector should be rotated. After this insight, Hamilton found a closed multiplication for four-dimensional complex numbers of the form $\mathbf{i}x + \mathbf{i}y + \mathbf{i}z$, where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Hamilton dubbed his four-dimensional complex numbers $quaternions$. The parallel to ordinary complex numbers stems from the imaginary parts. A quaternion is a collection of four real parameters, of which the first is considered as a scalar and the other three as a vector in three-dimensional space. In addition, quaternion represents an algebra. We refer to [8] for all definitions related to quaternions. A quaternion is simply a fourtuple of real numbers, and can be written as $Q = [w, x, y, z]$. Next, quaternion has its norm such that $|\hat{Q}| = \sqrt{w^2 + x^2 + y^2 + z^2}$.

Quaternions are suitable to represent rotations. Every rotation can be thought of as a right handed rotation of $-2\pi < \theta < 2\pi$ radians about some axis. Let us represent the axis of rotation with a unit norm three element coordinate vector $\hat{a}$ We will associate with this rotation the unit quaternion

$$\hat{Q} = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \hat{a}\sin(\frac{\theta}{2}) \end{bmatrix}$$

For completeness, we mention here that the associated rotation matrix of a unit quaternion $[w, x, y, z]^t$ is

$$\begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix}$$

The first preprocessing method based on quaternions tries to convert input (unfolded) image into the form of a set of rotation matrices. Some quaternion filtering methods can be found in [16, 1, 7, 3]. In our case, more simple conversion has been used. The process of preprocessing runs in following steps:

1. Read an original image of IRIS.
2. Unfold the image.
3. Regard the RGB pixel values as a coordinates in 3D space.
4. Construct quaternions from RGB values.

5. Normalize quaternion.

6. Convert quaternions to the rotation matrices.

7. Save output image, so that pixels are replaced by rotation matrices.

3.2 Contour levels

Following paragraph briefly describes second preprocessing method based on color leveling. More precisely, this represent an extended method of contour detection based on marching-squares algorithm. This is a computer graphics algorithm that generates contour lines for a two-dimensional scalar field. It is similar to the three-dimensional marching cubes algorithm. In case of marching squares, a new virtual grid is built over input image (see the Figure 4 - blue grid). Every cell of such grid has four corners with values (c_{ij}, where i=row, j=column) corresponding to the image pixels. In our case, the input image is grayscaled, which means that cell values are represented by numbers from interval $< 0, 255 >$. After selecting a threshold value t, e.g. $t = 100$, all corners are marked as *inside* or *outside*, according to the comparison of their values with the threshold; e.g. if $(c_{ij} < t) \Rightarrow$ *inside*, otherwise *outside*. The Figure 5 shows all possible cases of evaluation of corners and appropriate isolines (contour), which represents a partial result of marching square algorithm. There is no place to describe marching square algorithm in more detail, that is way we refer to [10, 9, 14, 13] for more information on this method.

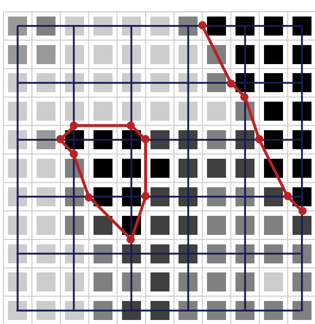

Figure 4. Image, marching squares grid and contours

The process of preprocessing runs in following steps:

1. Read an original image of IRIS.

2. Unfold the image.

3. Convert image into grayscale (8-bit per pixel).

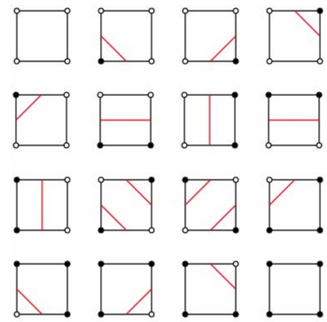

Figure 5. Marching squares - 16 cases

4. Find minimum and maximum value and normalize the image according to this interval.

5. Select the threshold for contour detection and get the set of levels

6. Run marching squares algorithm for every level, trace contours and fill regions with appropriate level color.

7. Save output image.

4 Singular Value Decomposition

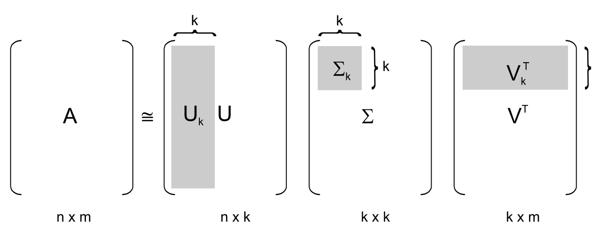

Figure 6. rank-k SVD

SVD [2] is an algebraic extension of classical vector model. It is similar to the PCA method, which has been the first method used for the generation of eigenfaces. Informally, SVD discovers significant properties and represents the images as linear combinations of the base vectors. Moreover, the base vectors are ordered according to their significance for the reconstructed image, which allows us to consider only the first k base vectors as important (the remaining ones are interpreted as "noise" and discarded). Furthermore, SVD is often referred to as more successful in recall when compared to querying whole image vectors [2].

Formally, we decompose the matrix of images A by *singular value decomposition* (*SVD*), calculating singular values and singular vectors of A.

We have matrix A, which is an $n \times m$ rank-r matrix and values $\sigma_1, \ldots, \sigma_r$ are calculated from eigenvalues of matrix AA^T as $\sigma_i = \sqrt{\lambda_i}$. Based on them, we can calculate column-orthonormal matrices $U = (u_1, \ldots, u_r)$ and $V = (v_1, \ldots, v_r)$, where $U^T U = I_n$ a $V^T V = I_m$, and a diagonal matrix $\Sigma = diag(\sigma_1, \ldots, \sigma_r)$, where $\sigma_i > 0, \sigma_i \geq \sigma_{i+1}$.

The decomposition

$$A = U\Sigma V^T$$

is called *singular decomposition* of matrix A and the numbers $\sigma_1, \ldots, \sigma_r$ are *singular values* of the matrix A. Columns of U (or V) are called *left* (or *right*) singular vectors of matrix A.

Now we have a decomposition of the original matrix of images A. We get r nonzero singular numbers, where r is the rank of the original matrix A. Because the singular values usually fall quickly, we can take only k greatest singular values with the corresponding singular vector coordinates and create a *k-reduced singular decomposition* of A.

Let us have k $(0 < k < r)$ and singular value decomposition of A

$$A = U\Sigma V^T \approx A_k = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

We call $A_k = U_k \Sigma_k V_k^T$ a k-reduced singular value decomposition (rank-k SVD) (U_0, Σ_0, and V_0 represent matrices filled with zeros).

Instead of the A_k matrix, a matrix of image vectors in reduced space $D_k = \Sigma_k V_k^T$ is used in SVD as the representation of image collection. The image vectors (columns in D_k) are now represented as points in k-dimensional space (the *feature-space*). For an illustration of rank-k SVD see Figure 6.

Rank-k SVD is the best rank-k approximation of the original matrix A. This means that any other decomposition will increase the approximation error, calculated as a sum of squares (*Frobenius norm*) of error matrix $B = A - A_k$. However, it does not implicate that we could not obtain better precision and recall values with a different approximation.

To execute a query Q in the reduced space, we create a reduced query vector $q_k = U_k^T q$ (another approach is to use a matrix $D'_k = V_k^T$ instead of D_k, and $q'_k = \Sigma_k^{-1} U_k^T q$). Instead of A against q, the matrix D_k against q_k (or q'_k) is evaluated.

Once computed, SVD reflects only the decomposition of original matrix of images. If several hundreds of images have to be added to existing decomposition (*folding-in*), the decomposition may become inaccurate. Because the recalculation of SVD is expensive, so it is impossible to recalculate SVD every time images are inserted. The *SVD-Updating* [2] is a partial solution, but since the error slightly increases with inserted images. If the updates happen frequently, the recalculation of SVD may be needed soon or later.

5 Quality evaluation

Since we need an universal evaluation of any retrieval method, we use some measures to determine quality of such method. In case of Information Retrieval we usually use two such measures - *precision* and *recall*.

In this case, however, we need to identify the correct person directly. In other words, we would prefer both the precision and recall to be as close to 1 as possible. Because of this, instead of calculating the precision at first match (which will be in this case identical with the precision at 100% recall), we have decided to adopt the binary approach (the person was/was not identified correctly). In the experimental section, we will present the percentage of people successfully identified by their irises for different methods.

6 IRIS collection

For testing of the different methods, we used a subset of a iris collection consisting of 384 irises. The iris were scanned by TOPCON optical device connected to the CCD Sony camera. The acquired digitized image is RGB of size 576×768 pixels. Some methods required us to represent each pixel by a single number. In such cases, only the red (R) component (near-infrared wavelengths) of the RGB image has been used in our experiments – it is known to be more reliable than recognition green or blue components or converting the irises to grayscale first [6].

The testing collection consisted of 128 irises of 64 people picked at random from the original collection. We also had an independent set of 10 query images.

An example of original (cropped) iris from the collection is shown in Figure 7.We did not isolate the central part and eyelids in this case to provide comparable results with [15].

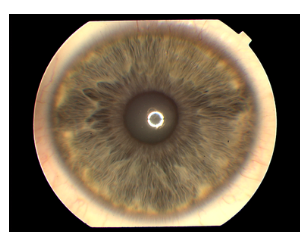

Figure 7. Original iris

The first modification was to normalize the collection in following way:

- Remove the parts outside of the iris from the image

- Convert the coordinate system from polar (clockwise) to Cartesian coordinates, independently on image dilation to create unfolded irises.

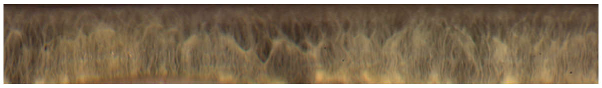

Figure 8. Unfolded iris

With this approach, we have obtained iris imaged of fixed size 1500×200 pixels.

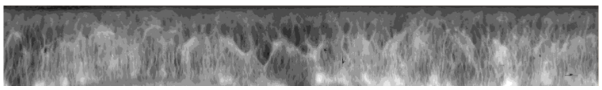

Figure 9. Iris after contour detection and color leveling

The use of quaternions and colour levels have been already described in section 3. Example of colour leveling result is shown in Figure 9. On the other hand, the quaternion matrix of given image could not be visualised this way.

7 Experimental Results

In this section, we will describe the results obtained on our collection.

7.1 Generated "Eigenirises" and Reconstructed Images

Many of tested methods were able to generate a set of base images, which could be considered to be "eigenirises" as is the case of PCA, SVD and several other methods. We are going to provide examples of both factors (base vectors) – "eigenirises" and reconstructed images which can be obtained from regenerated A_k (for original collection) where available.

We calculated results for all collection modifications in several dimensions, for the demonstration images we will use $k = 32$.

Because of SVD, we obtain factors with different generality, the most general being among the first. The first few are shown in Figure 10. The eigenirises with higher index bring more details to reconstructed images.

7.2 Query Evaluation

We have discarded the first coordinate of singular vectors from SVD, because it is tightly connected with the overall brightness of the image. Both methods improved the results of the LSI, however the improvement on our collection is quite small (see Table 1).

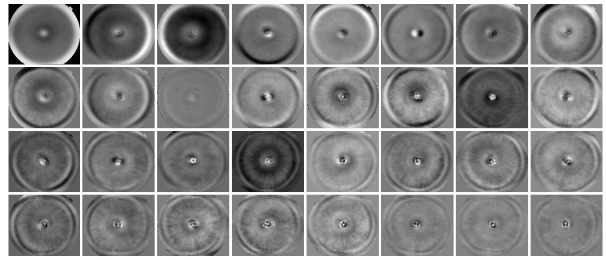

Figure 10. First 32 eigenirises for SVD method on original collection

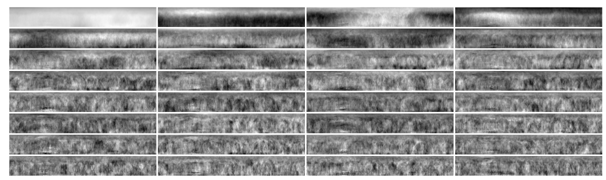

Figure 11. First 32 eigenirises for SVD method on unfolded collection

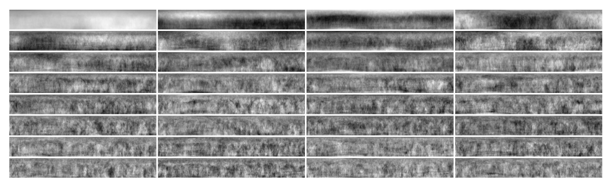

Figure 12. First 32 eigenirises for SVD method on colour-levelled collection

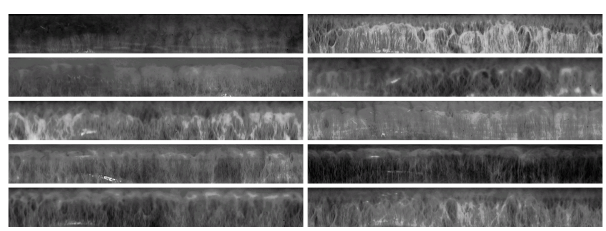

Figure 13. Irises used for querying (unfolded case)

Table 1. Query results after SVD - percentage of succesful identifications

k	Original	Unfolded	Quaternions	Contours
4	40%	50%	70%	30%
8	60%	90%	100%	90%
16	60%	90%	90%	100%
32	70%	90%	90%	100%
64	70%	100%	100%	100%

8 Conclusion

In this paper we presented a modified approach to contour and quaternion based IRIS preprocessing. This approach slightly improved the retrieval results with a small overhead for preprocessing which was done on GPU.

In future, we plan to use the Non-negative matrix factorization and other non-linear scaling functions to improve the results in lower-rank decompositions. Also, we are planing to test our approach on Casia iris collection, which contains irises captured in less suitable conditions and overall with lower resolution.

References

[1] D. Assefa, L. Mansinha, K. F. Tiampo, H. Rasmussen, and K. Abdella. Local quaternion fourier transform and color image texture analysis. *Signal Process.*, 90(6):1825–1835, 2010.

[2] M. Berry, S. Dumais, and T. Letsche. Computational Methods for Intelligent Information Access. In *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, San Diego, California, USA, 1995.

[3] T. Bulow and G. Sommer. Quaternion gabor filters for local structure classification. In *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition-Volume 1*, page 808, Washington, DC, USA, 1998. IEEE Computer Society.

[4] E. Dam, M. Koch, and M. Lillholm. Quaternions, interpolation and animation. Technical report, Department of Computer Science, University of Copenhagen, Denmark, 1998.

[5] J. Daugman. Statistical richness of visual phase information: Update on recognizing persons by iris patterns. *International Journal of Computer Vision*, 45(1):25–38, 2001.

[6] J. Daugman. The importance of being random: statistical principles of iris recognition. *Pattern Recognition*, 36(2):279–291, 2003.

[7] P. Denis, P. Carre, and C. Fernandez-Maloigne. Spatial and spectral quaternionic approaches for colour images. *Comput. Vis. Image Underst.*, 107(1-2):74–87, 2007.

[8] A. J. Hanson. *Visualizing Quaternions (The Morgan Kaufmann Series in Interactive 3D Technology)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[9] A. Lopes and K. Brodlie. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):16–29, 2003.

[10] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[11] P. Moravec, P. Gajdos, V. Snasel, and K. Saeed. Normalization Impact on SVD-based Iris Recognition. In *ICBAKE: 2009 INTERNATIONAL CONFERENCE ON BIOMETRICS AND KANSEI ENGINEERING*, pages 60–64. IEEE Comp Soc, IEEE, 2009.

[12] P. Moravec and V. Snasel. Dimension Reduction Methods for Iris Recognition. In Richta, K and Pokorny, J and Snasel, V, editor, *DATESO 2009 - DATABASES, TEXTS, SPECIFICATIONS, OBJECTS: PROCEEDINGS OF THE 9TH ANNUAL INTERNATIONAL WORKSHOP*, volume 471 of *CEUR Workshop Proceedings*, pages 80–89. CZECH TECHNICAL UNIV PRAGUE, 2009.

[13] G. M. Nielson. Dual marching cubes. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 489–496, Washington, DC, USA, 2004. IEEE Computer Society.

[14] G. M. Nielson, L. Zhang, K. Lee, and A. Huang. Parameterizing marching cubes isosurfaces with natural neighbor coordinates. In *GMP'08: Proceedings of the 5th international conference on Advances in geometric modeling and processing*, pages 315–328, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] P. Praks, L. Machala, and V. Snášel. Iris Recognition Using the SVD-Free Latent Semantic Indexing. In *MDM/KDD2004 – Fifth International Workshop on Multimedia Data Mining "Mining Integrated Media and Complex Data" in conjunction with KDD'2004 - The 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Section 2. Multimedia Data Mining: Techniques and Applications*, pages 67–71, Seattle, WA, USA, 2004.

[16] L. Shi. *Exploration in quaternion colour*. PhD thesis, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, 2005.

[17] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.