# Likelihood Ratio-Based Detection of Facial Features

Asker M. Bazen, Raymond N.J. Veldhuis, and Gerrie H. Croonen
University of Twente
Electrical Engineering, Mathematics, and Computer Science - Signals and Systems
P.O. box 217, 7500 AE Enschede, The Netherlands
Phone: +31 53 489 3156 Fax: +31 53 489 1060
E-mail: a.m.bazen@el.utwente.nl

*Abstract*— **One of the first steps in face recognition, after image acquisition, is registration. A simple but effective technique of registration is to align facial features, such as eyes, nose and mouth, as well as possible to a standard face. This requires an accurate automatic estimate of the locations of those features. This contribution proposes a method for estimating the locations of facial features based on likelihood ratio-based detection. A postprocessing step that evaluates the topology of the facial features is added to reduce the number of false detections. Although the individual detectors only have a reasonable performance (equal error rates range from 3.3% for the eyes to 1.0% for the nose), the positions of the facial features are estimated correctly in 95% of the face images.**

*Keywords*—**Face recognition, facial feature detection, likelihood ratio, registration.**

## I. Introduction

The task of biometric systems is to recognize persons from measurements of body characteristics. Face recognition is one of the most user friendly types of biometrics, since it can be applied transparently: without requiring specific user actions. Although there are many commercial face recognition systems available, and in spite of all academic research on the topic, the performance of current face recognition systems is still unsatisfactory.

A first step in face recognition, after image acquisition, is registration. Registration is the process in which a face image is aligned with an image of a standard face by means of a geometric transformation. Its purpose is to avoid recognition errors due to differences is size and position, caused by differences in camera positions and head poses.

A simple but effective technique of registration is to determine the transformation by aligning facial features, such as eyes, nose and mouth, as well as possible with the facial features in the image of the standard face. This requires an accurate automatic estimate of the locations of those features. This contribution proposes a method for estimating the locations of facial features based on likelihood ratio-based [1], [2] detection. The facial features (both positions and gray-scale appearance) can also be used as a starting point for face recognition.

The likelihood ratio-based facial feature detection algorithm is an alternative to the commonly used template matching. Under the assumption of Gaussian distributions, likelihood ratio detectors for both eyes, the nose and the mouth, are derived from a large set of unregistered faces taken from the FERET database [3]. The manually labelled facial feature positions that are provided with the database are used to determine a probability-density function of the relative positions of these facial features.

The locations of these facial features are determined as follows. First, for each facial feature, the likelihood ratio is computed at every pixel of the image. The locations with a likelihood ratio above a predetermined threshold are marked as candidate locations for the particular feature. This results in a set of candidate locations for each feature. The threshold is chosen such that the true location is very likely to be in the set.

Second, postprocessing is applied to reduce the number of false detections. For this purpose, the likelihood of the topology (relative positions) of all facial feature combinations is computed, and the combination with the highest likelihood is selected.

Although the individual detectors only have a reasonable performance (equal error rates range from 3.3% for the eyes to 1.0% for the nose), the positions of the facial features are estimated correctly in 95% of the images after postprocessing.

This paper is organized as follows. The proposed facial feature detection algorithm consists of two steps. Section II discusses the actual feature detection, while the postprocessing is explained in Section III. Experimental results from applying the proposed detection algorithm to a part of the FERET database are given in Section IV. Finally, Section V concludes the paper and gives some recommendations.

## II. Feature Detection

The common way to detect an object in an image is called template matching [4]. This method computes the

Fig. 1. Examples of a few face images as found in the FERET database.

position of the maximum similarity of the object (which is also called the template) to the image by shifting it pixel wise over the image, and computing the Euclidean distance at each position. The object is detected at a certain position if the similarity measure exceeds some predefined threshold.

If the image is normalized in energy, template matching can be implemented by a 2-dimensional cross correlation. In that case the template is applied as a 2-dimensional filter to the image. This approach can be regarded as the matched filter. It can be implemented even more efficient in the frequency domain, using fast Fourier transforms (FFTs).

The matched filter provides the optimal solution if two conditions are fulfilled. First, the object to be detected should be invariant, which means that the template is exactly equal to the object as it appears in the image. Second, the object should be embedded in additive white Gaussian noise. Although these conditions are not met most of the times, template matching is in many applications the method of choice. This is motivated by its computational efficiency.

The likelihood ratio-based feature detection that is presented in this paper differs from standard template matching by modelling the variations of the template and the background, instead of following the strict assumptions of template matching.

Our facial feature detection algorithm consists of two stages. First, the detector is trained using examples from facial feature templates and non facial feature templates. In the second stage, these variations are used to detect the facial features in new face images more reliably.

Feature detection is performed by means of a likelihood ratio, which results in optimal detection [1]. The likelihood ratio $L(\mathbf{x})$ of a template being a facial feature is given by:

$$L(\mathbf{x}) = \frac{p(\mathbf{x}|w)}{p(\mathbf{x}|\overline{w})} \tag{1}$$

where class $w$ represents the specific facial feature that has to be detected, $\mathbf{x}$ is a feature vector that represents the template to be classified, $p(\mathbf{x}|w)$ is the probability density of $\mathbf{x}$, given $\mathbf{x}$ is a member of class $w$, and $p(\mathbf{x}|\overline{w})$ is the probability density of $\mathbf{x}$, given $\mathbf{x}$ is not a member of class $w$.

Instead of testing against the non facial feature distribution, we can also test against the distribution of all templates, which gives a slightly different likelihood ratio to:

$$L(\mathbf{x}) = \frac{p(\mathbf{x}|w)}{p(\mathbf{x})} \tag{2}$$

where $p(\mathbf{x})$ is the prior distribution of all templates. In this framework, a test feature vector $\mathbf{x}$ is accepted as a facial feature template if its likelihood ratio exceeds a threshold $t \in [0, \infty\rangle$.

In this work, each template, consisting of $d$ pixels, is directly used as a $d$-dimensional feature vector, or a point in a $d$-dimensional space. The variation of the appearance of a template $\mathbf{x}$ around its mean $\mu$ is modelled by a multidimensional Gaussian probability density function:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \cdot |\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right) \tag{3}$$

where $\Sigma$ is the covariance matrix that represents the variations.

There are several motivations to choose a Gaussian probability density function. First, the pixels are measurements from some random physical process, and are there-

Fig. 2. Examples of a few right eye templates as found in the FERET database.

fore likely to be Gaussian distributed. This is even more likely after a linear dimension reduction that is introduced later on. Second, it is common practice to choose a Gaussian probability density function if the true distribution is not known and the number of examples is limited. The third reason for using Gaussian probability density functions is that it results in relatively simple processing in the detection phase.

Since the goal is to distinguish templates that contain facial features from all other templates, we construct two data sets. The first data set contains templates that are centered with respect to the hand-labelled facial feature coordinates. A few examples of the extracted right eye templates are shown in Figure 2. The second data set contains templates that are chosen at random positions from the face images. Once the data sets have been constructed, direct estimation of the mean and the covariance matrices is a straightforward task.

For each of the templates, we use the following notations. The class center $\mu_{\mathrm{W}}$ is the mean of all examples of the facial feature. The within-class covariance matrix $\Sigma_{\mathrm{W}}$ represents the differences between multiple templates of the facial feature. This includes variations due to scaling, rotation, lighting conditions, and between-person variations. The mean $\mu_{\mathrm{T}}$ is the mean of all randomly chosen templates. The total covariance matrix $\Sigma_{\mathrm{T}}$ represents the variations over all randomly chosen templates.

Given these parameters of the Gaussian distribution, the likelihood ratio that was defined in Expression 2 is given by:

$$\begin{aligned} L(\mathbf{x}) = & \frac{|\Sigma_{\mathrm{T}}|^{1/2}}{|\Sigma_{\mathrm{W}}|^{1/2}} \cdot \\ & \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_{\mathrm{W}})^T \Sigma_{\mathrm{W}}^{-1}(\mathbf{x}-\mu_{\mathrm{W}})\right. \\ & \left. + \frac{1}{2}(\mathbf{x}-\mu_{\mathrm{T}})^T \Sigma_{\mathrm{T}}^{-1}(\mathbf{x}-\mu_{\mathrm{T}})\right) \end{aligned} \quad (4)$$

By incorporating the values of the constants into the threshold, and using the log-likelihood ratio $\Lambda(\mathbf{x})$, the similarity measure $S(\mathbf{x})$ to be tested is given by:

$$\begin{aligned} S(\mathbf{x}) = & -(\mathbf{x}-\mu_{\mathrm{W}})^T \Sigma_{\mathrm{W}}^{-1}(\mathbf{x}-\mu_{\mathrm{W}}) \\ & + (\mathbf{x}-\mu_{\mathrm{T}})^T \Sigma_{\mathrm{T}}^{-1}(\mathbf{x}-\mu_{\mathrm{T}}) \end{aligned} \quad (5)$$

Expression 5 cannot be evaluated directly because of the inversion of the covariance matrices. If the number of examples is smaller than the dimension of the feature vector, the covariance matrix is singular and its inverse does not exist. This is referred to as the small sample size problem [5]. On the other hand, if the number of examples is not much larger than the dimension of the feature vector, the inverse of the covariance matrix will be very inaccurate, and the estimate from the training set will not be representative for the test set.

To deal with this problem, we reduce the dimension of the feature vector in two steps. First, the dimension is reduced by means of a principal component analysis (PCA) on $\Sigma_{\mathrm{T}}$. The effect of this transform is that the templates can be reconstructed with minimum squared error, given the reduced dimension [6]. The dimension reduction has the side effect of noise reduction, since the most important variations are maintained, while the other variations are discarded. Next, the dimension is further reduced by Fisher's linear discriminant analysis (LDA) [7], [8]. The result is that only those projections where the difference between both classes is largest are maintained, which further improves the recognition robustness.

The final step is a simultaneous diagonalization of the covariance matrices, which makes $\Sigma_{\mathrm{W}}$ identity and $\Sigma_{\mathrm{T}}$ diagonal [5]. The sequence of transformations described above can be replaced with one matrix multiplication by $\mathbf{V}$. Let $\nu_{\mathrm{W}} = \mathbf{V}\mu_{\mathrm{W}}$ and $\nu_{\mathrm{T}} = \mathbf{V}\mu_{\mathrm{T}}$ denote the transformed means, and let $\mathbf{y} = \mathbf{V}\mathbf{x}$ denote the transformed input feature vectors, then (5) reduces to:

$$\begin{aligned} S(\mathbf{y}) = & -(\mathbf{y}-\nu_{\mathrm{W}})^T(\mathbf{y}-\nu_{\mathrm{W}}) \\ & + (\mathbf{y}-\nu_{\mathrm{T}})^T \Lambda_{\mathrm{T}}^{-1}(\mathbf{y}-\nu_{\mathrm{T}}) \end{aligned} \quad (6)$$

Because $\Lambda_{\mathrm{T}}$ is a diagonal matrix of much smaller dimensions than the original covariances matrices, the number of computations has decreased considerably. While the direct application of Expression 5 would take $2d^2 + 2d$ multiplications, testing of the diagonalized reduced feature vector takes only $dr + 3r$ multiplications, where $r$ is the

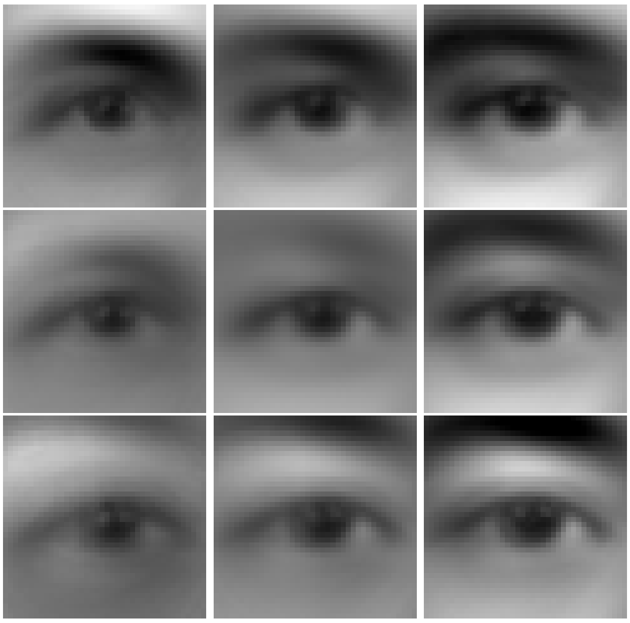

Fig. 3. The mean right eye template and its main variations. The change in horizontal direction gives the third important variation, while the change in vertical direction represents the second important variation.

reduced dimension. This is a reduction of a factor $2d/r$ of the number of multiplications. A more elaborate description of likelihood ratios and dimension reduction can be found in [9].

Figure 3 shows the main variations of the right eye templates as found in the FERET database. This visualizes the interpretation of likelihood ratio-based feature detection as deformable template matching [10]. In this figure, the most important variation is not shown, since it only affects the overall brightness of the template.

It is worth mentioning that selection of the optimum template size is less critical for likelihood ratio-based detection than it is in case of standard template matching. A template that is too large includes additional non-informative information, which decreases the detection performance due to the curse of dimensionality in case of standard template matching. On the other hand, likelihood ratio-based detection will recognize the information as non-informative, and therefore exclude it for detection purposes.

## III. Postprocessing

In order to reduce the number of false detections, a postprocessing step is added. This step does not consider the individual detections, but evaluates the topology, or relative positions, of the detected facial features. For this purpose, each combination of features is processed individually. In case of 5 candidates for all four features, this gives $5^4 = 625$ topologies to be evaluated.

The most simple evaluation of a given topology is to apply some hard constraints. For instance, the mouth should be located below the nose, and the left eye should be located to the right of the right eye, etc. However, this procedure does not reduce the number of valid candidates sufficiently, although it can be used as a first selection criterium.

A more sophisticated method makes use of a so called statistical shape model [11]. In this model, a shape is described by the topology of a set of landmark points. The $x$- and $y$-coordinates are taken as features and the mean and covariance matrix are estimated from a training set, after which whitening and dimension reduction is applied to isolate the main modes of variation. To constrain the variations maximally, all shapes are registered to the mean shape by means of a least-squares similarity transformation [12] that consists of translation, rotation and scaling. A new shape is registered to the mean shape, after which the likelihood of the shape can be calculated.

In this work, the statistical shape model approach has been applied as postprocessing. As landmarks, we take the facial feature locations. The model has been trained on the hand-labelled facial feature $x$- and $y$-coordinates.

Although the dimension of the feature vector is small compared to the number of examples, the least-squares similarity registration makes it necessary to apply dimension reduction. Applying the optimal translation (in $x$ and $y$) and the optimal rotation reduces the number of degrees of freedom of the feature vector by three, while the scaling projects the remaining variation on a part of a hypersphere. More details can be found in Appendix A. To account for these effects, PCA has been used to reduce the dimension of the feature vector from 8 to 4.

Once the statistical shape model has been trained, the likelihood of each candidate topology can be evaluated. In the current implementation, the most likely candidate is selected as final decision.

## IV. Experiments

We have trained and tested the facial feature detection algorithm on the FERET database [3]. From this database, we used a subset of 1603 frontal face images. The size of these images is 384 by 256 pixels, and the pixels contain 8-bit gray scale values. As illustrated in Figure 1, the faces show large variations in pose and lighting conditions.

For the extraction of training and test templates of the facial features, we used the ground truth information that is included with the database. This information contains the manually labelled coordinates of the eyes, nose and mouth. The same information has been used to evaluate

TABLE I
TEMPLATE PARAMETERS AND DETECTION PERFORMANCE

| | template size | dimension | EER |
|---|---|---|---|
| left eye | 50 × 50 | 12 | 3.3% |
| right eye | 50 × 50 | 12 | 2.3% |
| nose | 50 × 50 | 18 | 1.0% |
| mouth | 60 × 40 | 35 | 1.3% |

the performance of our facial feature detection algorithm.

In the FERET ground truth information, all facial features are labelled on their spatial center. Although this is fine for the positions of the eyes and the mouth, it gives some troubles for the the nose. The variation of the nose template is smaller if the noses are registered at the tip of the nose instead of at the (fairly arbitrary) middle of the nose. Therefore, we have adjusted the nose coordinates to the tip of the nose.

### A. Feature Detection

Although there are many more optimizations possible, we will present some preliminary results in this section. The detection performance of the individual facial features can be evaluated in a simple classification context. In that case, new examples of facial features and of non facial features are presented to the likelihood ratio-based classifier, and the performance is measured in terms of error rates. Table I summarizes the optimum template parameters, and the resulting equal error rates (EER).

When applying the likelihood ratio-based facial feature detector to an image, and selecting the largest likelihood ratio value, the correct facial features are found in 45% of the cases. When considering 5 candidates for each feature, the correct features are included in the candidate set in 80% of the cases. The performance increases to 95% if 10 candidates are considered.

The eyes are the most difficult facial features to detect correctly. Missed detections are most of the times caused by reflections into the subjects glasses. Glasses cause no troubles if the eyes can be seen through them.

Figure 4 shows examples of two faces and the corresponding likelihood ratios and facial feature detections for the right eye. The upper row shows a typical good result, where the true right eye position shows the largest similarity. The lower row shows a situation where the detector fails: higher similarity values for the right eye are found in the moustache and in the hair.

### B. Postprocessing

The postprocessing performs as expected. If 10 candidates per feature are used, the likelihood of the topology is evaluated, and additional constraints on rotation and scaling are applied, the postprocessing correctly selects the correct configuration of candidates. This means that our algorithm is able to detect the facial features in 95% of all tested face images.

Fig. 4. Examples of two faces and the corresponding likelihood ratios and facial feature detections for the right eye. The upper row shows a typical good result, where the true right eye position shows the largest similarity. The lower row shows a situation where the detector fails: higher similarity values for the right eye are found in the moustache and in the hair.

### C. Processing time

Virtually all processing time is spent in the actual feature detection step, since this involves large matrix multiplications. The processing time is linearly dependent on the size of the image, the size of the template, the decimation factor of image and template, the step size of shifting the template over the image, and the reduced dimension, with a unit time of $2.75 \cdot 10^{-9}$ s. In the setting that we used (image size is 256 by 384 pixels, template size is 50 by 50 pixels, no decimation, reduction to 12 dimensions, and a step size of 4 pixels), the processing time is equal to 430 ms per template on a 2.8 GHz Pentium-IV. This can be reduced by a factor 4 or 16 by applying a decimation with a factor 2 or 4 to the image and the template, resulting in processing times below 30 ms per feature.

## V. Conclusions and Recommendations

In this paper we have presented a new method for facial feature detection. It uses likelihood ratios to include estimates of statistical variations of the template and the background into the detector. This type of deformable template matching is applicable in other areas than facial feature detection as well. Although the individual detectors only have a reasonable performance (equal error rates range between 3.3% for the eyes and 1.0% for the nose), the positions of the facial features are estimated correctly in 95% of the images after postprocessing.

Selection of the optimal template size is far less critical than it is with standard template matching. However, in likelihood ratio-based template matching, estimation of the covariance matrix is critical, due to small or limited sample size effects. A simple way of dealing with these effects is to use one training set to estimate the dimension reduction transformation, and another set to optimize the reduced dimension.

Although the processing time is not excessive in the current implementation (100 ms per feature on a 2.8 GHz P-IV), some speed optimizations are possible. One optimization is to use a multi-resolution approach, which first analyzes the image on a coarse level (using decimation and relatively large step sizes), and then on a fine level at the most interesting locations. This will reduce the processing time per feature to less than 30 ms.

Another optimization is found in the the way of transforming input templates to a reduced dimension. Instead of transforming an input feature vector by means of a matrix multiplication, this step can be implemented by a number of filters that are applied to the image (one for each of the elements in the reduced space). This can be implemented very efficiently in the frequency domain, using fast Fourier transforms, which reduces the number of computations following the same principles that are used in standard template matching.

In a future version of the facial feature extraction algorithm, selection of the most likely set of facial features will be improved further by combining information from both the likelihood of the topology and the likelihood of the individual feature. Furthermore, the topology and candidates can be optimized iteratively. Once the face image has been registered to a standard face, the facial features will show less variation, which makes more reliable candidate selection possible. Finally, it is useful to evaluate not only local optima as facial feature location candidates, but also optimize in their neighborhoods. A facial feature position that is not a local optimum might give a better topological fit, and therefore a better overall likelihood. Note the similarities to the active appearance models that are described in [13].

## Acknowledgement

Portions of the research in this paper use the FERET database of facial images collected under the FERET program [3].

## Appendix A. On registration

The purpose of registration in the context of face recognition is first to determine a similarity transform (a combination of translation, rotation and scaling) that maps an input image onto a reference image, and second to apply this transform. The transform discussed here is chosen such that it maps a set of two-dimensional spatial coordinates of facial features (eyes, nose and mouth) in the input image onto the corresponding coordinates in the reference image. The transform is chosen such that the mean-squared error of the transform applied to the coordinates is minimal. First, we will derive an expression for the optimal similarity transform, given the coordinates of the facial features in the input and the reference image. Second, we will show that the transformed coordinates show non-linear dependencies.

### A. The similarity transform

We will use complex variables to denote two-dimensional spatial coordinates. E.g. the coordinates $(x, y)$ are denoted as a single complex variable $z = x + \mathrm{i}y$. Let $x_i$ and $r_i$, $i = 1, \ldots, M$, denote the input and reference coordinates of the facial features in the complex domain. If eyes, nose and mouth are used, $M = 4$. The translation in the similarity transform is denoted by $t$, scaling and rotation by a multiplication with $s$. The optimal $t$ and $s$ are now found as

$$\hat{t}, \hat{s} = \arg\min_{t,s} \sum_i (sx_i + t - r_i)^2. \tag{7}$$

The minimum can be found by equating the derivatives w.r.t. $t$ and $s$ to zero. This results in:

$$\begin{aligned} \sum_i x_i s + t - r_i &= 0 \qquad (8) \\ \sum_i x_i^H (x_i s + t - r_i) &= 0, \qquad (9) \end{aligned}$$

with the superscript $^H$ denoting the complex conjugate transpose, or just the complex conjugate in case of a scalar. From (8) it follows that

$$\hat{t} = \frac{1}{M}\left(\sum_i r_i - \sum_i x_i \hat{s}\right). \tag{10}$$

Without loss of generality, we choose the $r_i$ such that $\sum_i r_i = 0$ and prior to registration we perform a translation to the input coordinates such that $\sum_i x_i = 0$. As a result, $\hat{t} = 0$ and

$$\hat{s} = \frac{1}{\sum_j \|\mathbf{x}_j\|^2} \sum_i x_i^H r_i. \tag{11}$$

The input coordinates are then registered as coordinates $u_i$, $i = 1, \ldots, M$, given by

$$u_i = \frac{1}{\sum_j \|\mathbf{x}_j\|^2} \left(\sum_j x_j^H r_j\right) x_i. \tag{12}$$

### B. Linear and non-linear dependencies

The subspace $\mathcal{U} \in \mathbb{C}^M$ of points $u_i$ that the $x_i$ shows linear and non-linear dependencies. First, because of the initial translation such that $\sum_i x_i = 0$, we have that

$$\sum_i u_i = 0. \tag{13}$$

Second, for an arbitrary set of points $x_i$, $i = 1, \ldots, M$, we can write $x_i = \alpha r_i + \beta z_i$, with $\sum_i z_i^H r_i = 0$, $\sum_i |r_i|^2 = 1$, $\sum_i |z_i|^2 = 1$ and $\alpha$ and $\beta$ complex scalars. These $x_i$ are mapped onto

$$u_i = \frac{|\alpha|^2 r_i + \alpha^H \beta_i z_i}{|\alpha|^2 + |\beta|^2}. \tag{14}$$

Because $\sum_i z_i^H r_i = 0$ and $|\alpha|^2$ is a real scalar, no $x_i$, $i = 1, \ldots, M$, are mapped onto $\mathrm{i} r_i$, $i = 1, \ldots, M$. Or

$$\{\gamma \mathrm{i} r_i\} \notin \mathcal{U}, \tag{15}$$

for any real $\gamma \neq 0$. If we regard $\mathcal{U}$ as a subspace of $\mathbb{R}^{2M}$, (13) and (15) imply that $\mathcal{U}$ only has $2M - 3$ linearly independent dimensions.

In addition to the linear dependencies (13) and (15) there is a non-linear dependency in $\mathcal{U}$. By a straightforward calculation in follows for the $u_i$ that

$$\sum_i \|u_i - \frac{1}{2} r_i\|^2 = \frac{1}{4} \sum_i \|r_i\|^2. \tag{16}$$

This means that all points in $\mathcal{U}$ have a distance $\frac{1}{2}\sqrt{\sum_i \|r_i\|^2}$ to the point $\frac{1}{2} r_i$, $i = 1, \ldots, M$.

## References


[1] H.L. Van Trees, *Detection, estimation, and modulation theory*, Wiley, New York, 1968.

[2] A.M. Bazen and R.N.J. Veldhuis, "Likelihood ratio-based biometric verification," in *Proc. ProRISC 2002, 13th Annual Workshop on Circuits, Systems and Signal Processing*, Veldhoven, The Netherlands, Nov. 2002, pp. 181–189.

[3] P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The feret database and evaluation procedure for face recognition algorithms," *Image and Vision Computing J*, vol. 16, no. 5, pp. 295–306, 1998.

[4] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[5] S. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Inc., San Diego, CA, 1990.

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[7] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 711–720, July 1997.

[8] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Wiley-Interscience, 2nd edition edition, Oct. 2000.

[9] J.A. Kauffman, A.M. Bazen, S.H. Gerez, and R.N.J. Veldhuis, "Grip-pattern recognition for smart guns," in *Proc. ProRISC 2003, 14th Annual Workshop on Circuits, Systems and Signal Processing*, Veldhoven, The Netherlands, Nov. 2003.

[10] A.K. Jain, Y. Zhong, and M.P. Dubuisson-Jolly, "Deformable template models: A review," *Signal Processing*, vol. 71, no. 2, Dec. 1998.

[11] T.F. Cootes and C.J. Taylor, "Statistical models of appearance for computer vision," Tech. Rep., University of Manchester, Manchester, UK, Oct. 2001.

[12] S.H. Chang, F.H. Cheng, W.H. Hsu, and G.Z. Wu, "Fast algorithm for point pattern matching: Invariant to translations rotations and scale changes," *Pattern Recognition*, vol. 30, no. 2, pp. 311–320, 1997.

[13] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 23, no. 6, pp. 681–685, June 2001.