# Parabola-based Face Recognition and Tracking

Francis Deboeverie, Peter Veelaert and Wilfried Philips

***Abstract*—Several applications such as access control, behaviour observation and videoconferencing require a real-time method for face recognition and tracking. We propose to represent faces with parabola segments with an algorithm that allows us to fit parabola segments real-time to edge pixels. Parabola offer a good description for the many edges in the object, which is advantageous for several applications. We use parabola segments for face recognition and tracking, which is done with the parabola distance function, that matches parabola segments based on distance and intensity. We propose a global matching cost for face recognition and a technique to estimate motion vectors for face tracking by statistical analysis of the parabola distance function.**

## I. INTRODUCTION

Image analysis for human faces has been an active research topic in image processing, computer vision and psychology. Specific research areas include face detection, face tracking, face recognition, face animation, etc. In this paper, we focus on face recognition and tracking that can serve as a front end to other facial image analysis tasks, such as recognition of the facial expression and facial orientation.

Faces are similar in structure and show only small differences from person to person. Aspects as the differences in lighting, the large variety of distinct face expressions, the orientation and relative position of the face, changes in hairstyle or the presence of glasses make the analysis of faces still more complicated. The difference in appearance of a single face due to pose, expression and illumination are often larger than the differences between two faces of two different persons under similar circumstances. In addition, efficient coding is needed to reduce the size of face model databases, which should contain a single description for each face [1], [2]. Another desirable property, becoming prominent in recent applications, is that the recognition and coding algorithms can also be used to solve more general problems such as the recognition of pose and face expression, occluded or imprecisley localized faces, gender recognition or age estimation [3].

To continue improving individual features, the interrelationship between features can be exploited either by graph matching or by introducing global distance functions for feature matching. Lades et al. [5] presented a dynamic link structure for distortion invariant object recognition which employs elastic graph matching to find the closest stored graph. Thus dynamic link architecture is an extension of classical artificial neural networks. Memorized objects are represented by sparse graphs, its vertices are labeled with a multiresolution description in terms of a local power spectrum and the edges are labeled with geometrical distance vectors. Object recognition can be formulated as elastic graph matching, which is performed by stochastic optimization of a matching cost function. Wiskott and von der Malsburg [6] extended the technique. In general, the dynamic link architecture is superior to other face recognition techniques in terms of rotation invariant recognitions. Perronnin et al. [7] introduce a distance function which involves local transformations and consistency constraints for mapping Gabor features of the query image onto the model image. The drawback of these methods is that they are quite computationally expensive.

Humans recognize line drawings of faces as quickly and almost as accurately as gray-level pictures. Takacs [8] used edge maps to measure the similarity of face images. The faces were encoded into binary edge maps using the Sobel edge detection algorithm. The Hausdorff distance was chosen as a measure for the similarity of the two point sets, i.e., the edge maps of two faces, as it can be calculated without an explicit pairing of points in their respective data sets. The Hausdorff distance between point sets uses only the spatial information of an edge map without considering the inherent local structure and shape of the edges.

Gao and Leung [9] have successfully recognized faces by segmenting the edges into lines. The recognition system matches the lines of a Line Edge Map (LEM) of the query image with the LEM of the model image, using the Line Segment Hausdorff Distance. The LEM technique ensures that the sensitivity for noise and small changes in pose and expression is strongly reduced. LEM achieves a good recognition rate under pose and illuminations variations, with one model per person. The performance can degrade abruptly, however, when the face is occluded, rotated or the facial expression differs strongly from the expression stored in the database. Another disadvantage is that a line segment in itself does not represent a meaningful part of a face. To represent a part, e.g., an eyebrow, multiple line segments are needed.

This paper proposes several techniques to deal with some of the remaining shortcomings of LEM. In analogy with LEM, we propose a compact face feature, the Parabola Edge Map (PEM), which extracts parabola segments from a face edge map as features. This paper proposes parabola-based face coding and parabola matching techniques to integrate geometrical and structural features in the template matching.

The PEM approach not only has the advantages of geometric feature-based approaches, such as invariance for illumination and low memory requirement, but also has the advantage of high recognition performance of template matching. To represent a face we need far fewer parabola segments than line segments, which leads to larger more stable features. In fact, many parabola segments correspond to physically meaningful

F. Deboeverie (Francis.Deboeverie@ugent.be), P. Veelaert (Peter.Veelaert@hogent.be) and W.Philips(Philips@telin.ugent.be) are with the Department of Telecommunications and Information Processing(TELIN), Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium. Tel: +32 9 264 34 12, Fax: +32 9 264 42 95.

features such as eyebrows, cheekbones or lips. Furthermore, since a parabola segment has a more distinctive shape than a line segment, matching parabola segments is more reliable.

We extend the parabola distance function so that it also takes into account the intensity differences along a parabola segment. In fact, each parabola segment defines a convex region, so that it makes sense to compute the average intensity difference between the inner and outer side af a parabola. We will show that the recognition rate improves considerably by the introduction of intensity variations in the distance function.

We evaluate the PEM for all conditions of human face recognition, i.e., face recognition under controlled/ideal condition, varying lighting condition, varying facial expression, and varying pose. The system performances are compared to the LEM method. For the evaluation and comparison we use the Georgia Tech Face Database [10], The Database of Faces [11], the Bern University Face Database [12], The AR Face Database [13], the Yale University Face Database [14] and the Indian Face Database [15]. It is an encouraging finding that the proposed face recognition technique performs consistently superior to (or equally well as) the LEM and the eigenface method in all comparison experiments.

We also propose a real-time face tracking algorithm. Face tracking uses temporal correlation to locate human faces in a video sequence. With temporal information, we can narrow down the search range significantly and thus make real-time tracking possible. In this paper we follow a face by maintaining a bounding box which frames the face, and call this rectangle the face box. The face box is selected with human operator interaction in the first frame of the video sequence. We track the PEM of the face box in the video sequence by estimating motion vectors, which is done by statistical analysis of the parabola distance function. In several experiments the face is successfully tracked even in the presence of occlusion.

## II. Parabola segmentation of face edges

The proposed novel face feature representation, the Parabola Edge Map (PEM), integrates structural information and spatial information by grouping pixels of face edge map into parabola segments. To reduce noise, the grey-level image shown in Figure 1$(a)$, is preprocessed by a local average filter. Figure 1$(b)$ shows the edge features of a face are obtained with the Canny edge detector. The Canny edge detector is more suitable than the Sobel operator, because it results in thin edges of one pixel thickness and is less sensitive to noise. Digitized curves are obtained from the edge map by a simple boundary scan algorithm. To obtain parabola segments, we use our incremental linear-time fitting algorithm for curve segmentation which is based on constructive polynomial fitting [16], [17].

The output of the fitting algorithm is a list of parabola segments that approximate the edge map with an $L_\infty$ fitting cost, smaller than a user-specified threshold. The approximation is specified on a function $x(y)$ and $y(x)$, depending on which direction yields the smallest fitting cost. After segmentation the coefficients of each segment are optimized by quadratic regression.

$(a)$ $(b)$ $(c)$

Figure 1. **Canny edge detection:** Image $(a)$ shows a grey image of a face. Image $(b)$ shows the result after Canny edge detection. Image $(c)$ shows the complete face segmented in parabola.

PEM inherits or improves many of the nice properties of the line edge map. The PEM representation, which only records the end points and the coefficients of parabola segments, reduces the storage requirement. Between 20 and 50 parabola segments are sufficient to describe a face, with a storage cost of about 7 bytes per parabola segment. As illustrated in Figure 1$c$, PEM is a simple and natural description which still preserves sufficient information about the facial position and expression. PEM, as well as LEM, are expected to be less sensitive to illumination changes, because they are intermediate-level image representations derived from low-level edge map representation. In the next section we propose a method to match the parabola segments of different PEMs based on distance and intensity. The method makes use of the additional structural attributes of parabola segments.

## III. Shape Matching of Parabola Segments

Let $M = \{m_1, m_2, ..., m_f, ..., m_u\}$ and $T = \{t_1, t_2, ..., t_g, ..., t_v\}$ be two sets of parabola segments from different images, as shown on top in Figure 2. First, we define an asymmetrical distance function $d(m_f, t_g)$ for matching the parabola segment $m_f$ to the segment $t_g$. We propose a combined function of position, shape and intensity, which is largely independent for rotation, translation, scale, and global illumination. Parabola segments with main axis in the $x$-direction can be matched with parabola segments with main axis in the $x$-direction as well as with parabola segments with main axis in the $y$-direction and vice versa. Although this is not essential for the definition of $d(m_f, t_g)$, to facilitate the computation of the distance function, we subsample the parabola segments. We choose a collection of $n$ equidistant viewing points $Z = \{z_1, z_2, ..., z_n\}$ on the parabola segment $m_f$.

To obtain position, rotation and scale independency we compare $m_f$ with $t_g$ under different viewing angles, as shown in Figure 2. Let $\Psi = \{\psi_1, \psi_2, ..., \psi_r, \psi_\perp\}$ be a collection of $r + 1$ viewing orientations, where $\psi_1, \psi_2, ..., \psi_r$ denote

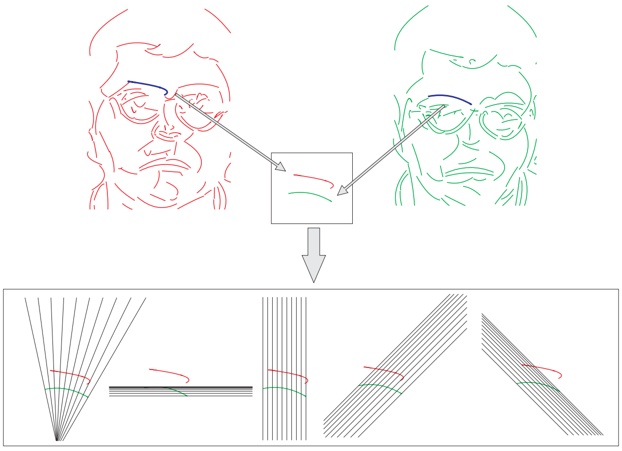

Figure 2. **The matching process of parabola segments:** On top: two sets of parabola segments. Below: comparison of two parabola under different viewing orientations $\Psi = \{\psi_\perp \, \psi_1, \psi_2, \psi_3, \psi_4\}$
.

$r$ viewing angles and $\psi_\perp$ denotes a varying viewing angle, perpendicular to the parabola $m_f$.

Let $\psi_i$ be one of the viewing angles. The following steps are then executed for each $\psi_i$. We determine the $s$ points of intersection $Q = \{q_1, q_2, ..., q_s\}$ of the straight lines with orientation $\psi_i$ through the points $z_j$ with the parabola segment $t_g$, note that $s \leq n$. That is, the intersection points are the points on $t_g$ as seen from the points $z_j$ in the direction $\psi_i$ . In this work we use $n = 10$ viewing points and we require that $s \geq 5$, which means that the parabola segments have to intersect with at least 50% of the lines. Otherwise, the distance becomes infinitely large. The process of building the matching function for the distance for a vertical viewing orientation is illustrated in Figure 3. The average distance is

$$d_A(m_f, t_g) = \frac{1}{s} \sum_{k=1}^{s} d_k, \tag{1}$$

where $d_k$ are the distances between the points of intersection $Q = \{q_1, q_2, ..., q_s\}$ and the viewing points $Z = \{z_1, z_2, ..., z_n\}$ measured along the viewing orientation. Let

$$\sigma_D^2(m_f, t_g) = \frac{1}{s} \sum_{k=1}^{s} (d_k - d_A)^2, \tag{2}$$

denote the variance of the distances.

The proposed distance function also takes into account how far the segments have been shifted relative to each other. Let $p_{m_1} = (x_1, y_1)$ and $p_{m_2} = (x_2, y_2)$ be the end points of $m_f$, and let $p_{t_1} = (x_3, y_3)$ and $p_{t_2} = (x_4, y_4)$ be the endpoints of $t_g$. The distances between the corresponding end-points of the parabola segments are measured perpendicular on the viewing orientation so they can be defined as

$$d_{P_1} = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} cos(\psi) \text{ and} \tag{3}$$

$$d_{P_2} = \sqrt{(x_2 - x_4)^2 + (y_2 - y_4)^2} cos(\psi). \tag{4}$$

The parallel distance is defined as

$$d_P(m_f, t_g) = \min(d_{P_1}, d_{P_2}). \tag{5}$$

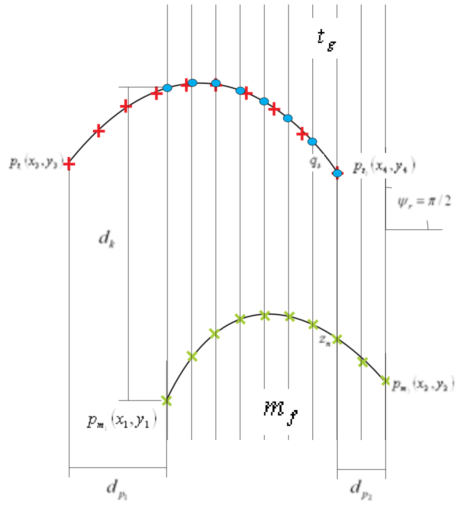

Figure 3. **The matching process based on distance for a vertical viewing orientation:** On the parabola $m_f$ there are selected ten viewing points. On the parabola $t_g$ eight points of intersection are shown. The distances are measured parallel with and perpendicular on the viewing orientation.

We define

$$D_\Psi(m_f, t_g) = \sqrt{(d_A * \sigma_{D_\psi})^2 + d_P^2}. \tag{6}$$

as a shape dissimilarity measure for two segments compared in the viewing direction $\psi$. By multiplying the average distance with the variance, we get a small value for $D_\Psi(m_f, t_g)$ when the parabola segments are parallel and translated, and a large value otherwise.

For the second, intensity dependent, part of the distance function, we consider the intensities above and below the parabola segment. One of the advantages of parabola segments over line segments is that parabola segments have a concave and convex side. We first introduce a dissimilarity measure for the convex side of the parabola segment. The comparison of intensities is illustrated in Figure 4. Let $\psi$ be a viewing direction. For each viewing point $z_n$ on the parabola segment $m_f$, we calculate an average value for the intensity at the convex side,

$$i^{conv}(m_f^k) = \sum_{j=1}^{v} w_j * i_j^{conv}(m_f^k) \text{ and } \sum_{j=1}^{v} w_j = 1, \tag{7}$$

where $i_j^{conv}(m_f^k)$ are the intensities (pixel values) at the convex side of the point $z_n$ along the viewing direction. In this work we used $v = 5$. The intensities are weighted, because the pixels closest to the parabola segments, and thus close to the edge map, can disturb the result. In this work we have used a gaussian kernel for the weights $w_j$. Similarly, for each intersection point $q_s$ on the parabola segment $t_g$ corresponding with $z_n$, we define

$$i^{conv}(t_g^k) = \sum_{j=1}^{v} w_j * i_j^{conv}(t_g^k), \tag{8}$$

where $i_j^{conv}(t_g^k)$ are the intensities (pixel values) at the convex side of the point $q_s$. The difference in average intensity at the

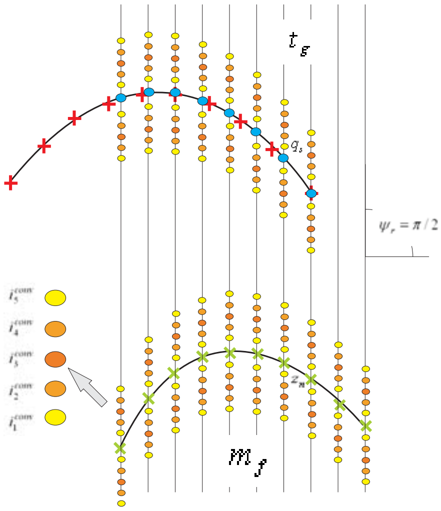


Figure 4. **The matching process based on intensity for a vertical viewing orientation:** On the parabola $m_f$ ten viewing points are selected. On the parabola $t_g$ eight points of intersection are shown. The pixels below and above the parabola segments which are taken into account for colour matching are shown as circles.

convex sides of the parabola segments for the corresponding points $z_n$ and $q_s$ is

$$i_k^{conv} = |i^{conv}(m_f^k) - i^{conv}(t_g^k)|. \tag{9}$$

The average difference in average intensity at the convex sides of the parabola segments is

$$i_A^{conv} = \frac{1}{s}\sum_{k=1}^{s} i_k^{conv}. \tag{10}$$

The variance for the differences in intensity is

$$\sigma^2_{A_{conv}} = \frac{1}{s}\sum_{k=1}^{s}(i_k^{conv} - i_A^{conv})^2. \tag{11}$$

Similar, we define $i_A^{conc}$ and $\sigma^2_{A_{conc}}$ at the concave side. The intensity dissimilarity function is defined as

$$I_\Psi(m_f, t_g) = \sqrt{(i_A^{conv} * \sigma_{A_{conv}})^2 + (i_A^{conc} * \sigma_{A_{conc}})^2}. \tag{12}$$

By multiplying the average differences in intensity with their variances, we reduce the value of the intensity function when the parabola segments have the same relative transition in intensity between their convex and concave side. Finally, the disparity between the segment $m_f$ and the segment $t_g$ along the direction $\psi$ is defined as

$$d_\psi(m_f, t_g) = \left(\frac{n}{s}\right)^2 \sqrt{D_\Psi(m_f, t_g)^2 + I_\Psi(m_f, t_g)^2}. \tag{13}$$

The division by the square number of intersection points $s$, reduces the disparity value when there are more intersection points. The (asymmetrical) distance between the parabola segments $m_f$ and $t_g$ is defined as the minimum disparity over all viewing directions originating from $m_f$,

$$C(m_f, t_g) = \min_\psi(d_\psi(m_f, t_g)). \tag{14}$$

The parabola segment $m_f$ matches with the parabola segment from the set $T$, for which the distance is minimal. The match is acceptable if this distance is below a certain threshold $A$, a typical value is ten:

$$C(m_f, T) = \min(C(m_f, t_g)) \leq A. \tag{15}$$

Let $M' \subseteq M$ be the set of parabola segments for which there are acceptable matches. The distance between the set of segments $M$ and the set of segments $T$ is defined as

$$S(M,T) = \frac{\sum_{f in M'}^{w}(l_{m_f} C(m_f, T))}{\sum_{f in M'}^{w} l_{m_f}} \left(\frac{|M| - |M'|}{|M|}\right), \tag{16}$$

where $l_{m_f}$ is the length of the segment $m_f$, and $|M|$ and $|M'|$ represent the size of the two sets. Large parabola segments have a larger weight than small segments. The distance for matching the segments $M$ with segments $T$ is made symmetrical by introducing the distance measure which will be used to compare faces:

$$H(M,T) = \max(S(M,T), S(T,M)). \tag{17}$$

### A. Face Recognition

The performance of the PEM method is compared to that of LEM. Four face databases were used in the experiments: the Georgia Tech Face Database [10], the Database of Faces [11], the Bern University Face Database [12], the AR Face Database [13]. For each person in a database we compare a face input to the face models of all other persons. A person is recognized correctly if the matching cost of its face input and the person's own face model is a minimum. An example for PEM is shown in Figure 5$(a)$, corresponding parabola segments are indicated with the same colour. The results of the LEM method are shown in Figure 5$(b)$. The results for testing an entire database for PEM and LEM are shown in Figure 5$(c)$ and Figure 5$(d)$, respectively. The distributions at the left side of each graph describe the density of the matching cost for matching faces of the same person. The distributions on the right describe the density of the matching costs for matching faces of different persons. For PEM the distributions are more separated, which indicates that the PEM method performs better than the LEM method.

Recognition results for all databases are shown in Table I. The second column shows the results for the LEM method. The third column shows the results using a combination of the distance and intensity function. The results are top1 classification, the correct match is only counted when the best matched face from a model is the correct person. We test face recognition under controlled condition and size variation, under varying lighting condition, under varying facial expression and under varying pose. It is found that the PEM approach performs better than the LEM method. We have an average increment of the recognition rate of 10.15% for face recognition under controlled condition, 36.41% for face recognition under size variation, 35.84% for face recognition under varying pose, 8.41% for face recognition under varying lighting and 29.12% for face recognition under varying facial expression.

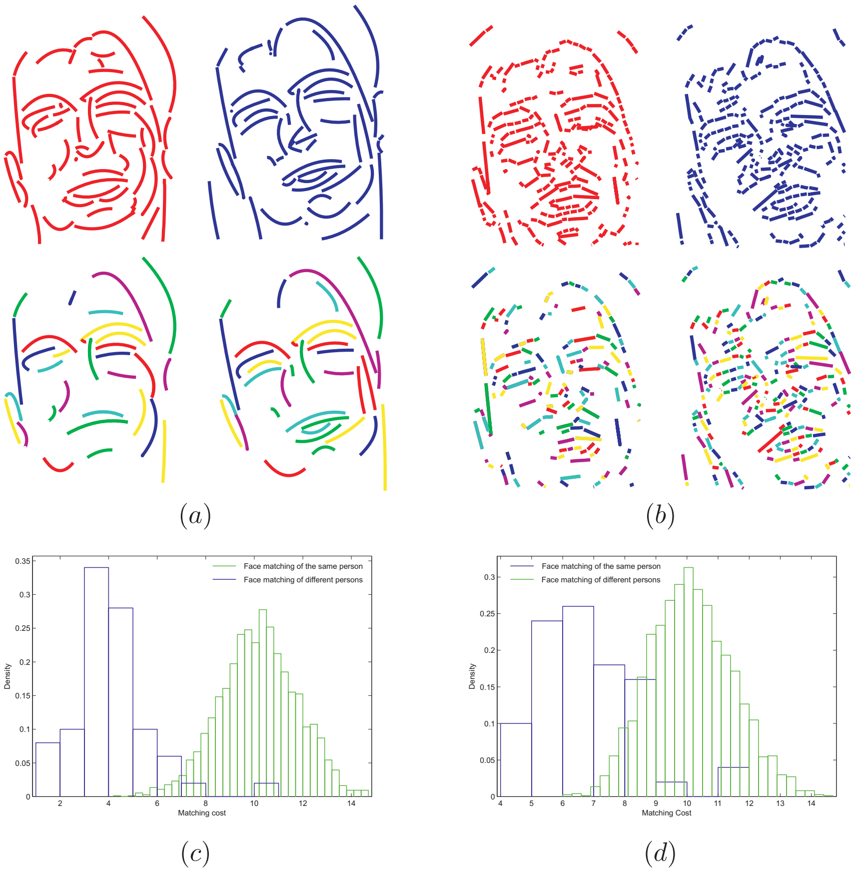


Figure 5. **Results for the matching process:** Image $(a)$ shows the parabola segments for two different faces of the same person and the matching results for the parabola segments, corresponding parabola segments are indicated with the same colour. Image $(b)$ shows the line segments for two different faces of the same person and the matching results for the line segments. Image $(c)$ and $(d)$ indicate the results in a graph for matching respectively with PEM and LEM for the whole database [10].

We assume that the main reason for the better results are the larger and more stable parabola segments, where each parabola segment describes a significant part of the face. Many parabola segments correspond to physically meaningful features. Since a parabola segment has a more distinctive shape than a line segment, matching parabola segments is more reliable. Furthermore, each parabola segment defines a convex region, so that it makes sense to compute the average intensity difference between the convex and concave side af a parabola. The recognition rate improves considerably by the introduction of intensity variations in the distance function.

| | LEM | PEM |
|---|---|---|
| *Controlled condition* | | |
| GTFD | 84,00% | **98,00%** |
| ATT | 95,00% | **100,00%** |
| BERN | 80,00% | **100,00%** |
| AR | 96,40% | **98%** |
| *Varying pose* | | |
| BERN Right | 55,00% | **93,34%** |
| BERN Left | 48,33% | **86,67%** |
| BERN Up | 46,67% | **86,67%** |
| BERN Down | 45,00% | **73,34%** |
| *Size variation* | | |
| AR with size variation | 53,80% | **90,21%** |
| *Varying lighting condition* | | |
| AR with left light on | 92,86% | **96,34%** |
| AR with right light on | 91,07% | **94,84%** |
| AR with both lights on | 74,11% | **92,10%** |
| *Varying facial expression* | | |
| AR with smiling expr. | 78,57% | **96,53%** |
| AR with angry expr. | 92,86% | **97,30%** |
| AR with screaming expr. | 31,25% | **96,21%** |

Table I
RESULTS FOR MATCHING: % CORRECTLY RECOGNIZED FACES.

### *B. Face Tracking*

We follow a face by maintaining a bounding box which frames the face, and call this rectangle the face box. In stead of face detection, we let the user demarcate the face with a face box in the first frame of the video sequence. The edges within the face box are segmented in parabola features (PEM). We search the face in a new incoming frame, by matching the parabola segments of the face box in the previous frame with the parabola segments in the current frame, using the parabola distance function.

The motion vector of the bounding box is estimated by statistical analysis of this parabola distance function. The global direction $\Psi_m$ of the motion vector, in which the face moved, is estimated from the histogram of the directions $(\psi_1, \psi_2, ..., \psi_r)$ in which the parabola segments individually moved. This histogram is shown in Figure $6(a)$. The length

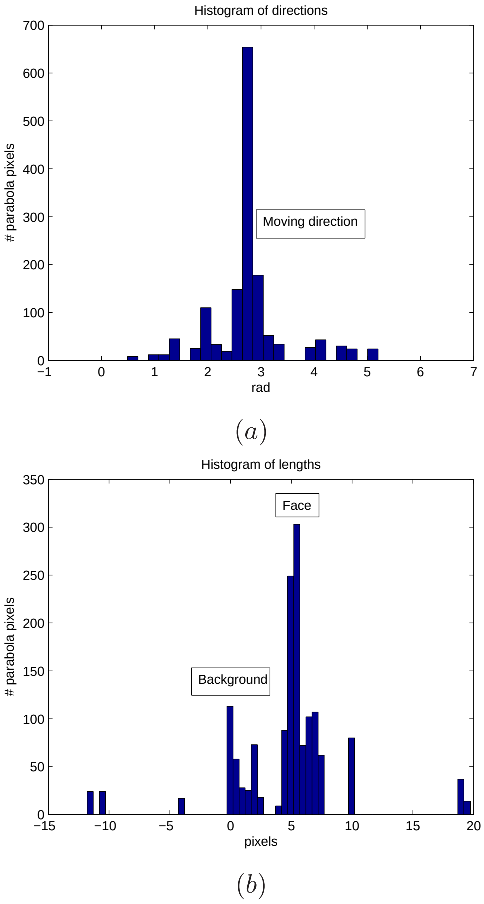


Figure 6. **Results for the matching process:** Image $(a)$ shows the histogram of the directions in which the parabola segments individually moved. Image $(b)$ shows the histogram of the distances in which the parabola segments individually moved, projected perpendicular on the axis in the earlier estimated direction of the motion vector.

$D_m$ of the motion vector is estimated from the histogram of the distances $d_A$ in which the parabola segments individually moved, projected perpendicular on the axis in the earlier estimated direction $\Psi_m$ of the motion vector. This histogram is shown in Figure 6$(b)$. The histograms take into account the lengths of the parabola segments.

The motion vector can be made more accurate. The previous PEM is moved to the current PEM according the motion vector. We calculate the mean distance between the moved previous PEM and the current PEM for shifted versions of the previous PEM. We keep the location that minimizes the mean distance between all corresponding parabola. The same is done for enlarged and reduced versions of the previous PEM. We now have a more accurate location and size of the face box in the new frame.

For the prediction of the motion vector, we can use more history. This is done by the use of the estimated directions and lengths of previous motion vectors. They are taken into account with different weight factors.

The results on different video sequences are promising, they prove the stability and robustness of parabola features. This kind of tracking is advantageous for solving problems with occlusion.

## IV. CONCLUSION

In this work we recognize and track faces with parabola features collected in a Parabola Edge Map (PEM). We presented a technique of shape matching for parabola segments, based on distance and intensity. We also presented a technique to estimate motion vectors for face tracking by statistical analysis of the parabola distance function. Parabola segments are robust describing low-level features, they are a good description for the many edges in a face, which is advantageous for several applications. We made a comparison with line features and achieve better results in experiments on different databases. In the future parabola features are useful for describing all kind of stable objects.

## REFERENCES


[1] Aleix M. Martinez, *Recognizing Imprecisely Localized, Partially Occluded, and Expression Variant Faces from a Single Sample per Class*, IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 6, june 2002.

[2] Tae-Kyun Kim and Josef Kittler, *Locally Linear Discriminant Analysis for Multimodally Distributed Classes for Face Recognition with a Single Model Image*, IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 3, march 2005.

[3] Hakan Cevikalp, Marian Neamtu, Mitch Wilkes and Atalay Barkana, *Discriminative Common Vectors for Face Recognition*, IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 1, january 2005.

[4] Ajmal S. Mian, Mohammed Bennamoun and Robyn Owens, *An Efficient Multimodel 2D-3D Hybrid Approach to Automatic Face Recognition*, IEEE transactions on pattern analysis and machine intelligence, vol. 29, no. 11, march 2007.

[5] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, and M. Konen, *Distortion Invariant Object Recognition in the Dynamic Link Architecture*, IEEE Trans. Computers, vol. 42, pp. 300-311, 1993.

[6] L. Wiskott and C. von der Malsburg, *Recognizing Faces by Dynamic Link Matching*, Neuroimage 4, pp. S14-S18, 1996.

[7] Florent Perronnin, Jean-Luc Dugelay and Kenneth Rose, *Fellow, A Probabilistic Model of Face Mapping with Local Transformations and Its Application to Person Recognition*, IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 7, july 2005.

[8] B. Takacs, *Comparing Face Images Using the Modified Hausdorff Distance*, Pattern Recognition, vol. 31, pp. 1873-1881, 1998.

[9] Yongsheng Gao and Maylor K.H. Leung, *Face Recognition Using Line Edge Map*, IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 6, june 2002.

[10] Georgia Institute of Technology, *Georgia Tech Face Database*, http://www.anefian.com/face_reco.htm.

[11] AT&T Laboratories, Cambridge, *The Database of Faces*, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[12] University of Bern, *Bern, Switzerland, Bern University Face Database*, ftp://iamftp.unibe.ch/pub/ImagesFaceImages/.

[13] A.M. Martinez and R. Benavente, *The AR Face Database*, CVC Technical Report #24, June 1998.

[14] University of Yale, *Yale University Face Database*, http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[15] Indian Institute of Technology Kanpur, *Indian Face Database*, http://vis-www.cs.umass.edu/%7Evidit/IndianFaceDatabase/.

[16] Peter Veelaert, *Constructive fitting and extraction of geometric primitives*, CVGIP: Graphical Models and Image Processing 59, 233–251, 1997

[17] Peter Veelaert and Kristof Teelen, *Fast polynomial segmentation of digitized curves*, Lecture Notes in Computer Science: Proc. Discrete Geometry for Computer Imagery, Vol. 4245, pp. 482-493, Szeged, Springer, 2006.