Profile Authentication Using a Chamfer Matching Algorithm

Stéphane Pigeon - Luc Vandendorpe
UCL - Laboratoire de Télécommunications et Télédétection
Place du Levant, 2 – B-1348 Louvain-La-Neuve – Belgium
E-Mail : pigeon@tele.ucl.ac.be

Abstract

This paper investigates the use of the profile shape to recognize human faces. In order to test the intrinsic efficiency of profile authentication, we will make use of a method that directly works on the profile contour encoded as x-y coordinates. This way, the performance rated here will only depend on the ability of the profile view to dissociate faces and not on the choice of a given set of profile features and/or the quality of their extraction. *Keywords : person verification, profile, Chamfer.*

Introduction

Profile authentication can be seen as an easy way to recognize people. Although we cannot expect as much identification potential from the profile shape as from other techniques like fingerprint analysis for example, profile recognition can be useful in the framework of a multi-modal person identification scheme together with other techniques like facial analysis or speech recognition [1].

On the other hand, the Chamfer matching technique searches for the best match between two binary images. Geometric transformations are used to distort one image (here referred as the *candidate image*) to another (the *reference image*) in order to minimize a given distance measure between them. These binary images are often derived from the image edges. Here, we make use of the shape of the profile. The fact that this technique works using the shape of the profile directly (no "feature" extraction required) is one of its advantages : the performance rated in this paper will only depend on the profile ability to dissociate faces and not on the choice of the possible features that could be extracted. Unfortunately, unlike feature based methods, handling the profile globally does not permit to adjust the relative importance of different areas of the profile.

The paper is organized as follows : first we will explain how the Chamfer algorithm works (section 1) and how it can be optimized (section 2). Then we will see how to extract normalized profile shapes from profile views (section 3). After having defined our testing setup (section 4), extensive results will be drawn in order to define profile's ability to dissociate faces (section 5).

1 The Chamfer Matching Algorithm

As the first step of the algorithm we generate a *distance map* from the reference profile. This distance map associates with each pixel of the reference picture, its distance from the closest profile pixel (all profile pixels get thus the zero distance value). As the true Euclidian distance is costly to compute, we use a *sequential Chamfer distance approximation* [2]. The sequential Chamfer distance algorithm starts from an zero/infinity image where each pixel is set to zero if it belongs to the profile, infinity otherwise. The distance map is obtained by applying the next formulas, from left to right/top to bottom first, and right to left/bottom to top afterwards (two passes are enough) :

```
for i = 2 to #rows - 1
for j = 2 to #cols - 1
  dist(i,j) = minimum{dist(i-1,j-1)+4, dist(i-1,j)+3, dist(i-1,j+1)+4, dist(i,j-1)+3, dist(i,j)}

for i = #rows-1 downto 2
for j = #cols-1 downto 2
  dist(i,j) = minimum{dist(i,j), dist(i,j+1)+3, dist(i+1,j-1)+4, dist(i+1,j)+3, dist(i+1,j+1)+4}
```

By superposing the candidate image on this distance map and by summing up all distances found along the candidate profile (the non-zero pixels in the candidate image), we get an estimate of the global distance that stands between them (mean squared criterion).

Actually, we cannot directly compare the reference and the candidate profiles together. The candidate profile has first to be compensated from the possible geometric transformations that can affect a face from one shot to the other, i.e. translations along x and y axes, scaling factor (related to the distance between the face and the camera) and rotation in the x/y plane (due to the possible rotation of the head). Given a set of values for these parameters, we then build a *compensated profile* from the candidate profile. This compensated profile is superposed on the reference distance map and a new global distance is computed. The best match between the candidate and the reference profiles is obtained by finding the set of parameters minimizing this global distance. It thus reverts to minimize a compensation function which, in our case, depends on the translation, scale and rotation variables. This minimization is done through a classic multidimensional minimization method. We make use of the *Downhill simplex algorithm*, which requires only function evaluations and no derivatives [4].

The global matching process is illustrated in figure 1. First, the candidate profile is projected onto the reference distance map and a global distance is computed. By minimizing this distance, the optimum compensation parameters are found (translation, scale and rotation). Then, the residual distance between the best compensated and the reference profiles is used to decide whether the two profiles belong to the same person or not.

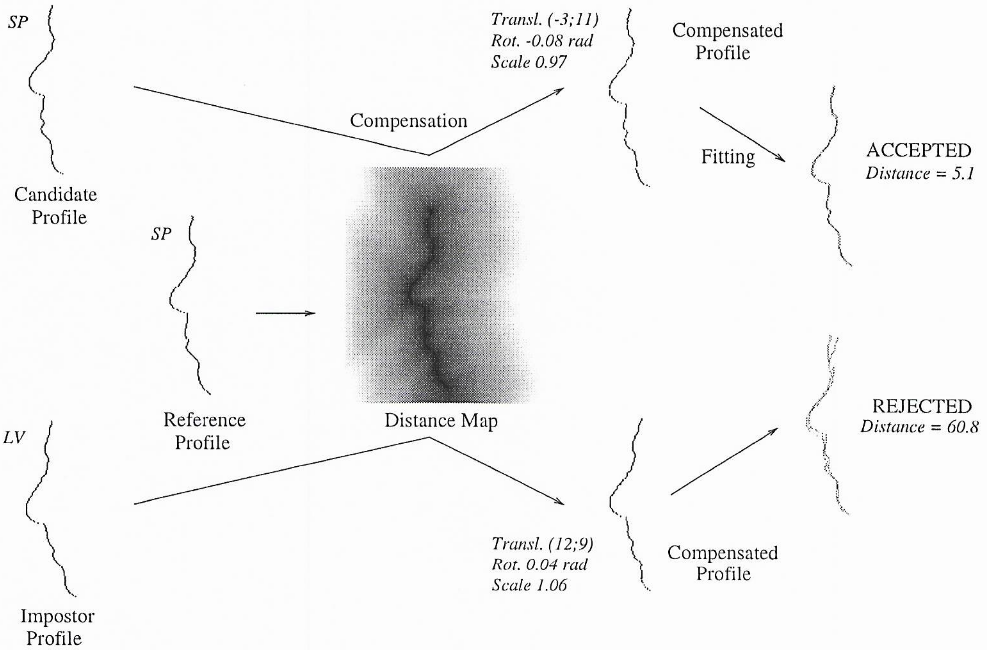


Figure 1: The Chamfer Matching Process

2 Optimization of the Chamfer/Simplex Algorithm

To avoid the simplex algorithm to converge towards a local minimum, attention must be paid to the initial parameters values used to initialize the algorithm. These values have to be as close as possible to the final solution in order to converge efficiently. The translation parameters along both x and y axes are estimated first by comparing the positions of nose's tip between reference and candidate profiles, the scale factor is found by comparing the two profile heights and the rotation parameter is arbitrary set to zero. These values are used to initialize a first Chamfer/simplex algorithm that runs into a low-resolution mode (candidate profile and reference distance map are down-sampled by a factor of 4 in both x/y directions).

As output, we get refined values for translation and scale parameters and a pretty good estimation for the head rotation. All these values are used to start the final full-resolution search.

Attention must be paid to the equations describing the geometric transformations between one profile and the other. In particular, we have to avoid as much as possible the influence of one parameter to the other(s). For example, by compensating the profile for rotation, we might have chosen a rotation center – usually the (0;0) coordinate – that is far from profile center of mass. On a theoretical viewpoint, it does not matter since every rigid transformation can be described by a rotation followed by a translation whatever the center that has been set for the rotation. In practice, chosing a rotation center that is far from the center of mass of the object to be rotated, induces an additional (and ofter large) translation of the object that has to be compensated for. In such a case, the simplex algorithm encounters some problems when being close to the minimum, but still searching for the best rotation parameter (same problem for the scale factor). Changing the rotation value will shift the profile and introduce an additional translation : then the translation vector (tx;ty) previously found to be correct is not valid anymore. We can solve this problem by properly centering the rotation and the scaling around the center of mass of the profile. By denoting *(tx;ty)* the translation parameters, *(cx;cy)* the profile's center of mass, *scale* the scale factor and *angle* the rotation angle, the following equations were finally used :

```
X' = (tx+r) + scale*[(X-cx)*cos{angle} + (Y-cy)*sin{angle}]
Y' = (ty+s) + scale*[(cx-X)*sin{angle} + (Y-cy)*cos{angle}]
```

At last, a problem occurs when the candidate profile extends over the reference profile : then, the lowest residual distance does not correspond to the best match anymore. To get rid of such a situation, we both match the candidate against the reference *and* the reference against the candidate. The lowest score is taken as the final score.

3 Profile Segmentation

The different profiles are automatically extracted from the M2VTS database profile views [5]. This database is made of 37 faces, provides 5 different shots for each face and offers a 286x350 pixel resolution. Profile views have a tolerance of about 90 degrees $\pm$ 15 degrees. Profile segmentation from profile views is divided into two steps : first, the head is segmented from the grey background, then, the profile is extracted from the head.

3.1 Head Segmentation

The face is segmented from the grey background by means of color clustering according to the method proposed in [3]. This segmentation can be summarized into the following steps :

1. The image is low-pass filtered in order to smooth the different color components and reduce the effect of noise inside the different color areas.

2. A 2-D color histogram is computed using normalized red R/(R+G+B) and green G/(R+G+B) components. As the background represents the largest surface in the picture, its location inside the histogram is given by the highest peak. The background can be fully segmented by connecting the color components lying around the background peak.

3. By backprojecting all pixels belonging to the background cluster into the original image, the background is extracted from the image (figure 2).

3.2 Profile Extraction

Once the head is segmented, the profile must be extracted from the head. In order to achieve good recognition results, we have to restrict the extraction to the invariant parts of the profile only, i.e. :

- do not include the forehead as part of the profile when it can be affected by the hair-style

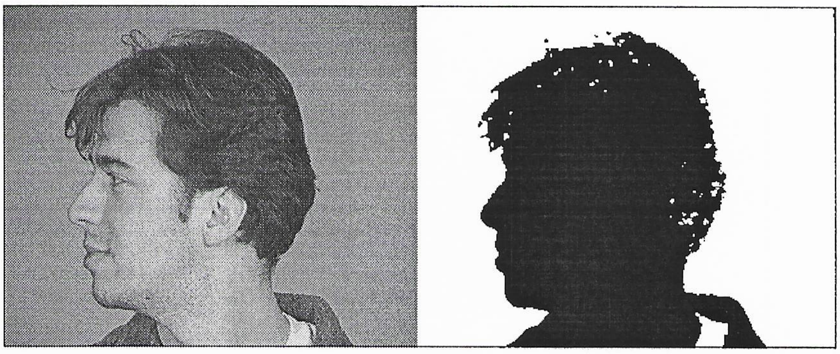


Figure 2: Result of the background segmentation by means of color clustering

- do not include the area below the chin, as its contour highly depends on the tilt of the head which is likely to change from one shot to another
- do not include the lower part of bearded faces

Depending on the face topology (hair-style, moustache, beard,...), we can choose between different profile extraction modes for each person of the database. A first mode selects the full profile and assumes short hair and the absence facial hair. A second mode only selects the lower part of the profile and has to be used when hair are suspected to cover the forehead. A third mode selects the upper area of the face (people wearing a moustache or a beard). The last mode is a combination of the two previous ones and selects the nose area only.

In order to achieve good results using the Chamfer matching algorithm, it is also important to select exactly the same part of the profile from one shot to the other. If the profiles are not normalized, it will be difficult to match them even if they belong to the same person, as a bias will be introduced in the residual Chamfer distance.

Taking into account the previous remarks, our profile extraction has been divided into the following steps (see figure 3) :

1. *Nose tip search.* The tip of the nose is located by searching for the leftmost point of the profile. Unfortunately, this point can be found in the hair area for some particular hair-styles and when the head is tilted downwards. Therefore, the maximum search will be restricted inside a window covering the middle part of the face.

2. *Location of the top of the nose.* From the tip of the nose, we move upwards along the profile up to the first local minimum. This minimum is considered as being the top of the nose.

3. *Rough profile segmentation.* From the nose tip and top, we compute the nose height. This height is taken as a reference in order to extract normalized profiles i.e. areas that extends over :

   - $[tip - 2 * height]$ to $[top + 0.7 * height]$ for the first mode (full profile)
   - $[tip - 2 * height]$ to $[top]$ for the second mode (no forehead)
   - $[tip - 0.3 * height]$ to $[top + 0.7 * height]$ for the third mode (no mouth/chin)
   - $[tip - 0.3 * height]$ to $[top]$ for the fourth mode

4. *Profile refinement.* For some faces, the chin is located above the $[tip - 2 * height]$ level. In those cases, the lower chin – and sometimes the shoulder – are part of the extracted profile. Lower chin and shoulder must be suppressed from the extracted profile as their shape is not invariant from one shot to the other. The lower chin can be located thanks to its almost horizontal orientation. The shoulder, that sometimes might overlap the chin, can be detected thanks to its particular orientation (from bottom left to top right). All pixels lying below the chin or belonging to the shoulder are removed from the profile.

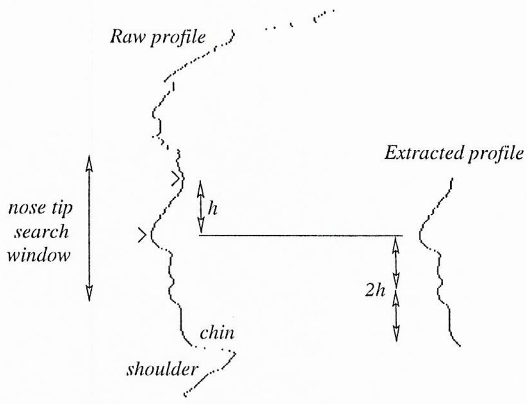


Figure 3: Profile extraction

4 Testing Setup and Training Session

4.1 Testing Setup

Four different shots taken from the M2VTS database are used to perform our tests. These shots were taken at one week intervals. Profiles taken from shots 1/2/3 are used to select the best *reference* profiles during a training session. Shot 4 provides the *candidate* profiles. Authentication results are computed by matching the different candidate profiles with their best references.

As each shot of the M2VTS database is made up of 37 faces, this testing setup allows 37 correct authentication matches and 37*36 intrusion matches. Moreover, this set of 37 faces is divided into two categories depending on the face topology. Faces for which we are able to extract full profiles are classified in the *full profile* class (24 profiles). The other profiles, which are partly extracted due to particular hair-style or the presence of a beard, are classified in the *partial profile* class (13 profiles).

4.2 Training Session

During the training session, three matching operations are performed for each profile : the first shot (S1) is matched with shot 3 (S3), S2 with S1 and S3 with S2. For example, if the matchings S1/S3 and S3/S2 lead to the two lowest scores (i.e. the lowest residual Chamfer distances), then S3 will be taken as the reference shot for that profile as it appears in both best matchings.

5 Results

Three experiments have been conducted :

- The first experiment matches S4 profiles with the best reference profiles issued from the training session. Figure 4 shows the recognition (R), false rejection (FR=100-R) and false acceptance (FA) curves as a function of a threshold. A given match is considered to be correct if its score (residual Chamfer distance) is located below the threshold, rejected otherwise.
- The second experiment matches the S4 profiles with each S1, S2 and S3 profiles (aggregated reference) and takes
$$min\{score(S4,S1), score(S4,S2), score(S4,S3)\}$$
as a global score. Figure 5 shows the same R/FR/FA curves as a function of the threshold. This second experiment achieves improved results than experiment 1 but is more CPU demanding.
- The third experiment studies the influence of wearing glasses on the profile recognition. The M2VTS database is restricted to faces wearing glasses only (13 faces). Reference profiles are issued from

the same training session as for the first experiment (glasses off during the training session). Figure 6a shows authentication results when people of S4 are asked to put their glasses off. Figure 6b shows these same results when people are allowed to keep their glasses. The relatively good results of figure 6 can be explained by the fact that the glasses colors are close to the color of the gray background and are thus automatically extracted from the profile during the segmentation process.

6 Conclusion

The following table summarizes the main results given above and provides the different false acceptance rates when the recognition rate is set to 85%.

	Global Score	Full Profiles Score	Partial Profiles Score
Trained reference	6.9%	1.1%	9.5%
Aggregated reference	5.7%	1.0%	8.1%

Table 1: False acceptance rates achieved for a given recognition rate of 85%

These figures validate the recognition scheme being used here. Profile analysis is able to achieve good results but they depend on the features that can be extracted from the profile : as seen in table 1, the false acceptance scores computed on partial profiles is about one order of magnitude higher than for the full profiles. We could get rid of this problem by asking the people to clear their forehead in front of the camera.

Another way to take into account this disparity of results would be to deliver, instead of a unique final score (the residual Chamfer distance between the two profiles), two different measures : the final *Chamfer matching score*, providing a good idea on how close to each other the reference and the candidate profiles are (this score should be as small as possible) and the *length of the profile* being used to match the two profiles, interpreted as a confidence measure of the first score (the highest possible). Indeed, the smaller the profile, the less confident you are in the matching *validity*, as the number of candidate profiles successfully matched with the reference increases when the comparison is performed on a small part of the profile only.

Acknowledgments

This work has been performed within the framework of the M2VTS Project (*Multi Modal Verification for Teleservices and Security applications*) granted by the European ACTS programme.

References

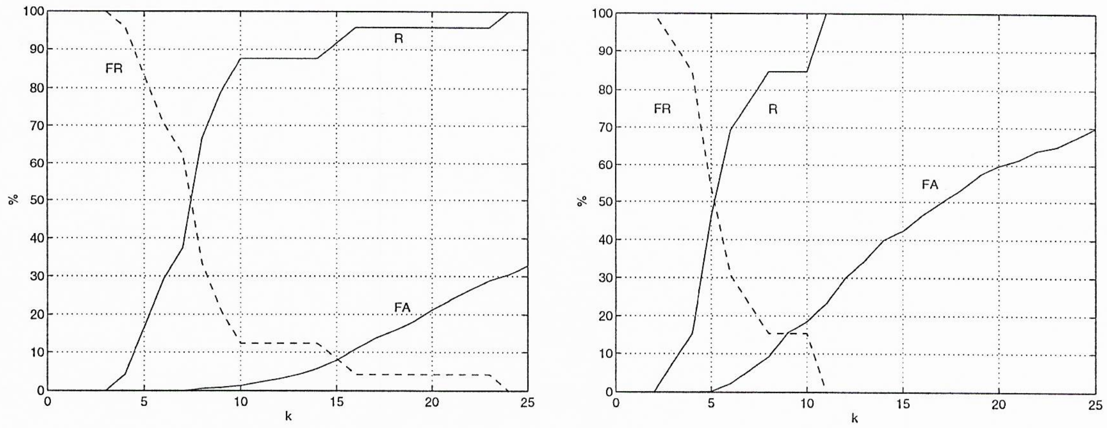
[1] M. Acheroy, C. Beumier, J. Bigün, G. Chollet, B. Duc, S. Fischer, D. Genoud, P. Lockwood, G. Maitre, S. Pigeon, I. Pitas, K. Sobottka and L. Vandendorpe. "Multi-Modal Person Verification Tools using Speech and Images". In *Proceedings of the European Conference on Multimedia Applications, Services and Techniques (ECMAST '96)*, pp. 747-761, Louvain-La-Neuve, May 28-30 1996.

[2] Gunilla Borgefors, "Hierarchical Chamfer Matching : A Parametric Edge Matching Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 849-865, Vol. 10, No. 6, November 1988.

[3] J. Matas and J. Kittler. "Spatial and Feature Space Clustering: Applications in Image Analysis". In *Proceedings of the 6th Int. Conf. on Computer Analysis and Patterns*, Prague, September 6-8 1995.

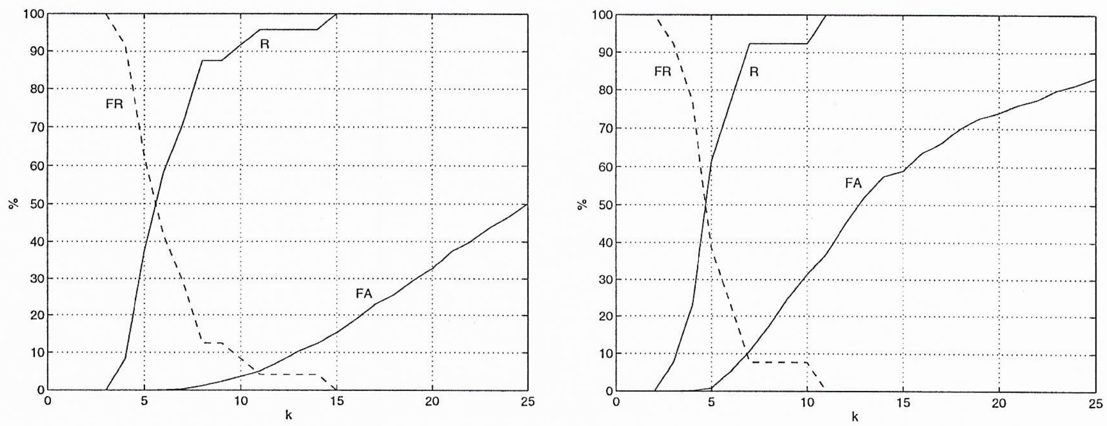
[4] W. H. Press, S. A. Teukolsky and W. T. Vertterling, *Numerical Recipes in C : The Art of Scientific Computing*, Cambridge University Press, 1988.

[5] The M2VTS Database (Release 1.00) : see <http://www.tele.ucl.ac.be/M2VTS/>



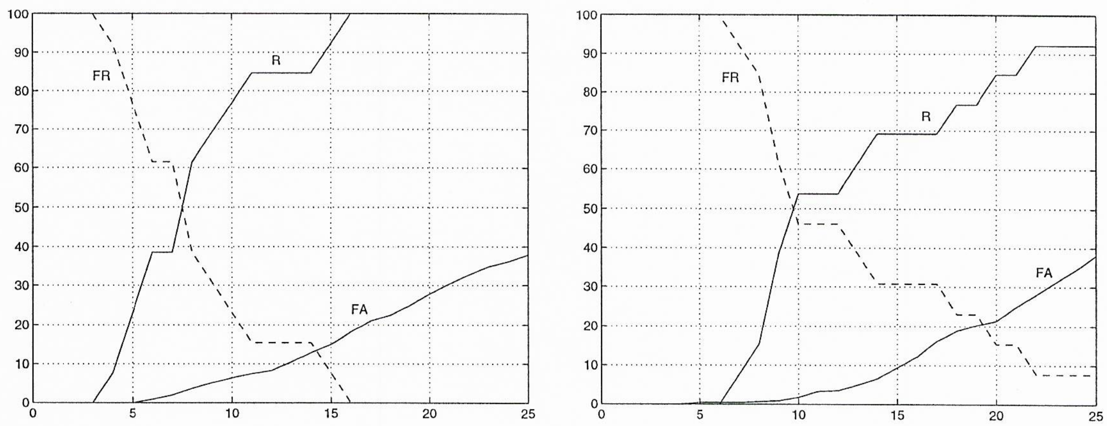
(a) Full profiles (b) Partial Profiles

Figure 4: Matching against trained reference



(a) Full profiles (b) Partial Profiles

Figure 5: Matching against aggregated reference



(a) Glasses off (b) Glasses on

Figure 6: Matching against trained reference : influence of glasses