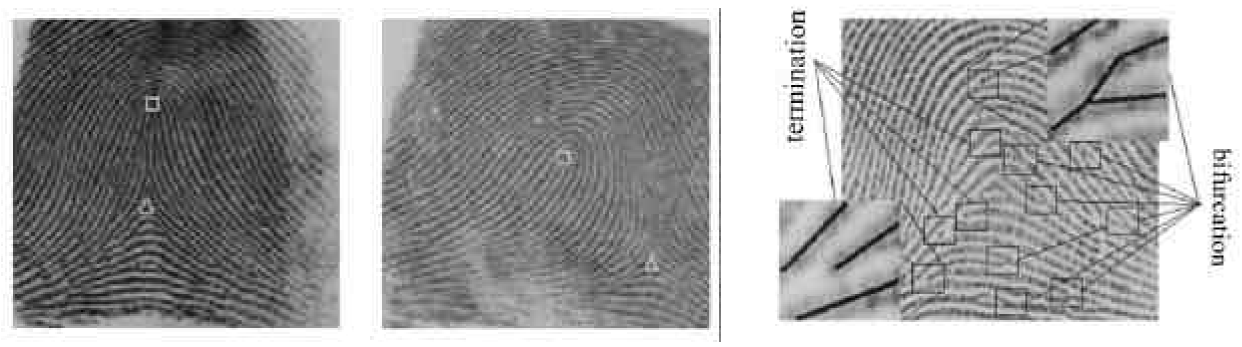
Fingerprint recognition

Fingerprint image enhancement, classification with neural networks and matching
Extract of Bachelor Thesis

The application of biometrics in personal identification caused a fast evolution of this field. Biometrics offers a wide scale of possibilities for identification, from the iris scan to the body odor analysis. The most reliable and widely used technique is the fingerprint analysis. It has already been used in criminology, for almost a hundred years. Nowadays its usage has been extended to commercial tasks, such as personal identification for on-line banking, security systems, etc.

The identification is being performed using the local and global fingerprint features. We call global features some properties of the ridge system, such as pattern, density and the presence of two specific formations, the core and delta points. Local features are the ridge termination and bifurcation. These are the basis of fingerprint matching.



delta, core and minutiae points

Every automatized fingerprint identification system has to be able to complete certain tasks, such as the fingerprint image enhancement, feature extraction, classification and matching. Our application is not different, we use the same steps to solve the most important problems: classification, the 1:1 match (determining whether two impressions belong to the same finger or not) and the 1:N match (searching for a finger in a database). Fingerprint classification is being made on the basis of the global features (ridge pattern). It was developed to ease the search in classic (not automatized) databases. In automatized systems, it can be used to optimize the search, reducing the database necessary to search.

image enhancement

As we already mentioned, the first step in fingerprint identification is the image enhancement. The most common method for acquiring an impression of a finger is the classic ”ink on paper” method, which is also the most unreliable. These types of fingerprint images may contain many typical errors; blur regions (where the finger slides) different intensities in different regions and so on. These errors need to be corrected, in order to get a reliable feature set. There are many different approaches; we used the most advanced one, the direct grayscale image enhancement using Gabor filters, proposed by Anil K. Jain (1). We slightly modified the original method, adding a preliminary step, the histogram equalization, and changing the orientation determination process. The enhancement consists of 5 steps (as shown below): histogram equalization, normalization, orientation image computation, frequency image computation and filtering.

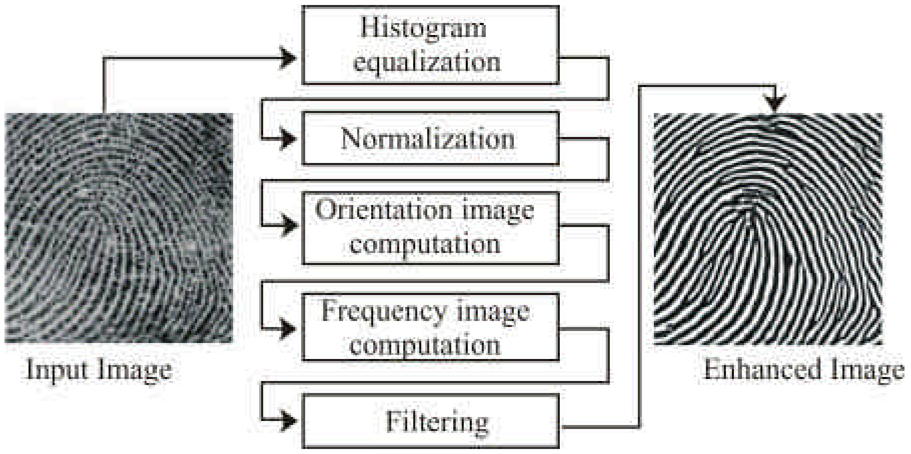


Image enhancement process

The first would be the histogram equalization, which is performed in local windows, so the differences between regions with different intensities are corrected. The normalization converts the equalized image, into a normalized one, which means that the image gets a predefined mean and variance of pixel intensities. We have to determine the local orientation and density of the ridges, in order to adjust the adaptive Gabor filter in every region.

The orientation image computation is based on the method proposed by Maio & Maltoni (2). The orientation angle in a local neighborhood is the local dominant direction of ridges. We divide the image into squared windows, the so-called tangent windows. In every (i_h, j_k) pixel of the tangent window, the $n_{hk} = [a_{hk}, b_{hk}, 1]$ vector is computed, as follows:

$a_1 = T(i_{h+1}, j_{k+1}), a_2 = T(i_{h-1}, j_{k+1}), a_3 = T(i_{h-1}, j_{k-1})$ and $a_4 = T(i_{h+1}, j_{k-1})$

$$a_{hk} = \frac{-a_1 + a_2 + a_3 - a_4}{4}, \qquad b_{hk} = \frac{-a_1 - a_2 + a_3 + a_4}{4}$$

The $t = (t_1, t_2)$ vector (which represents the requested direction) can be computed as follows:

$$A = \sum_{\substack{h=1..a\\k=1..a}} (a_{hk})^2, \quad B = \sum_{\substack{h=1..a\\k=1..a}} (b_{hk})^2, \quad C = \sum_{\substack{h=1..a\\k=1..a}} a_{hk} b_{hk}$$

$$t = \begin{cases} \left[1,\ \frac{B-A}{2C} - \operatorname{sgn}(C)\sqrt{\left(\frac{B-A}{2C}\right)^2 + 1}\right] \text{if } \mathrm{C} \neq 0 \\ [1,0] \ \text{ if } \mathrm{C} = 0,\ \mathrm{A} \leq \mathrm{B} \\ [0,1] \ \text{ if } \mathrm{C} = 0,\ \mathrm{A} > \mathrm{B} \end{cases}$$

The requested j_0 direction is:

$$j_0 = \begin{cases} \arctan\left(\frac{t_2}{t_1}\right) & \text{ha } \mathrm{t}_1 \neq 0 \\ \frac{p}{2} & \text{különben} \end{cases}$$

So in every pixel of the tangent window, the orientation parameter would be j_0.

The frequency image is computed using the already determined orientation parameters. Once again the image is divided into $w \times w$ Ý16 × 16Þ blocks, and for every block, an oriented window of size $l \times w$ Ý32 × 16Þ is calculated. For each block, we compute the x-signature, Xß0à, Xß1à, ...Xßl ? 1à where:

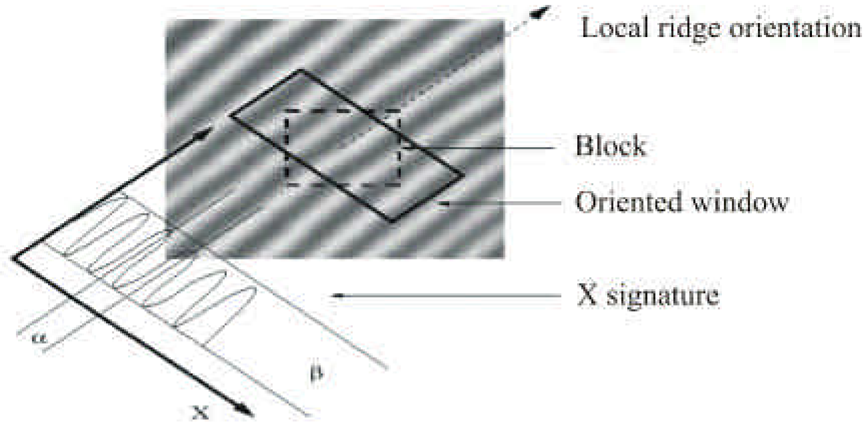
$$X[k] = \frac{1}{w}\sum_{d=0}^{w-1} G(u,v), \quad \mathrm{k} = 0,1,..\mathrm{l}\text{-}1$$

$$u = i + \left(d - \frac{w}{2}\right)\cos O(i,j) + \left(k - \frac{l}{2}\right)\sin O(i,j),$$

$$v = j + \left(d - \frac{w}{2}\right)\sin O(i,j) + \left(\frac{l}{2} - k\right)\sin O(i,j)$$

Where *G* is the normalized input image, *O* is the orientation image.

In regions, where the image is not corrupted, and no minutiae points appear, the x-signature forms a discrete sinusoidal-shape wave. The frequency of ridges can be estimated from the x-signature, as the average number of pixels between two consecutive peaks. In those blocks, where the frequency cannot be estimated correctly, it needs to be interpolated from the neighboring blocks. After the interpolation a low-pass filter is used to smooth the frequency image.

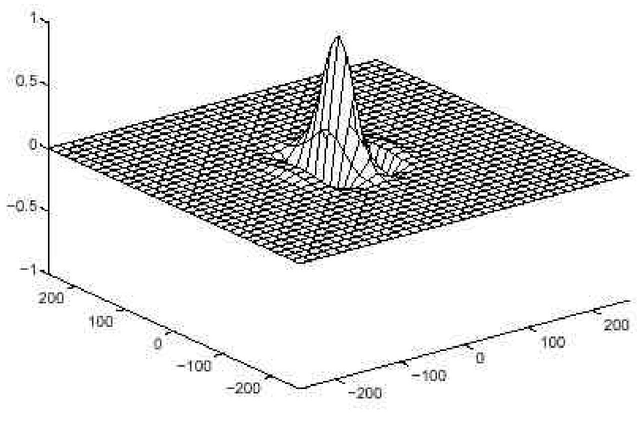


Oriented window and the calculation of the X signature

The last step of the enhancement process is the filtering. The general form of the even symmetric Gabor filter is

$$h(x, y : f, f) = \exp\left\{-\frac{1}{2}\left[\frac{(x\cos f)^2}{d_x^2} + \frac{(y\sin f)^2}{d_y^2}\right]\right\}\cos(2pfx\cos f)$$

Where d is the orientation, f the frequency parameter, N_x, N_y are constants.



even-symmetric Gabor filter

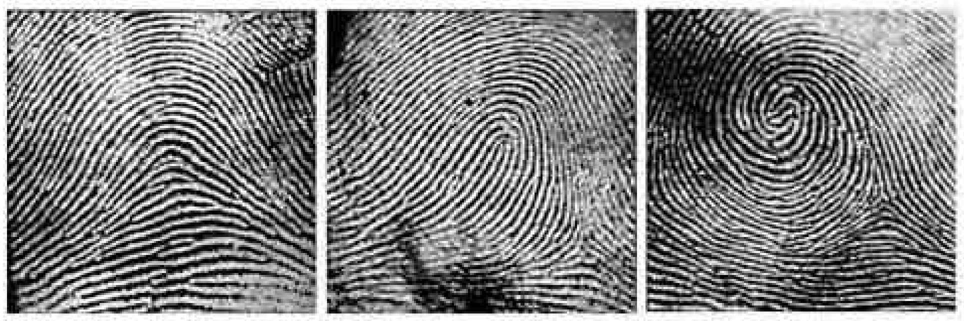
The E filtered image is computed as follows:

$$E(i, j) = \sum_{u=-w_g/2}^{w_g/2}\sum_{v=-w_g/2}^{w_g/2} h(u, v : O(i, j), F(i, j))G(i-u, j-v)$$

Where O is the orientation image, F the frequency image and G is the normalized image.

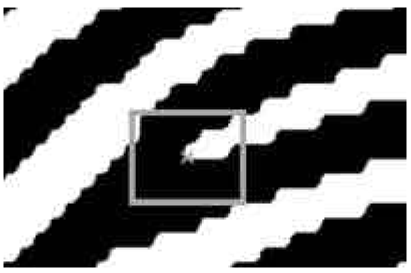
Classification

In our application, the classification distinguishes 4 classes, the arch, left and right loop and the whorl. We used a two layer MLP (multilayer perceptron). This type of neural networks has been successfully used in pattern recognition, so we considered it acceptable for this task. The MLP consists of four layers: an input layer with 400 input neurons, two hidden layers, each containing 25 neurons, and an output layer, with 4 neurons, each corresponding to one class. The input vector is built from the orientation image, which is simplified into a 20 × 20 matrix then converted into a vector.


Fingerprint classes: arch, loop, whorl

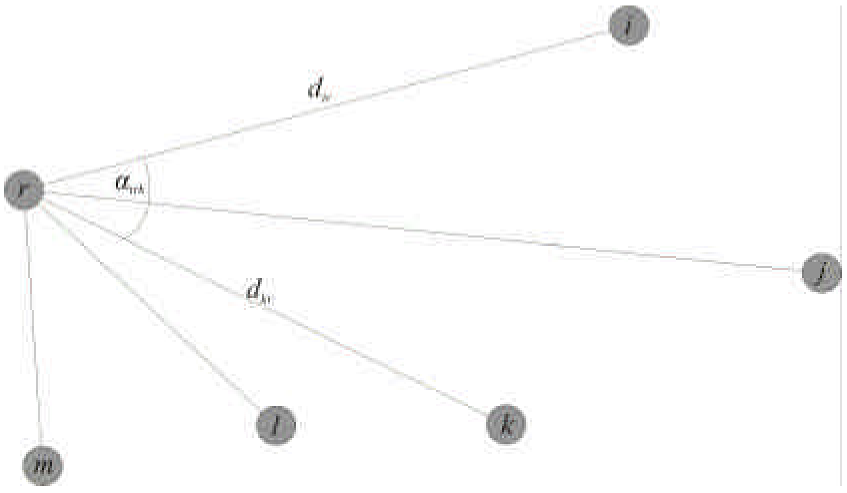
Matching

Matching is based on the local features (minutiae). These can be detected in many ways. The most common technique is the thinning, when ridges are thinned to one pixel wide lines. But during the thinning, the minutiae may change their location, so we developed another method, without thinning. We search for minutiae in local windows, considering the surroundings. After the filtering, the image is binarized or in other words converted into a so-called binary ridge image, where every black pixel belongs to a ridge, every white pixel belongs to a valley. The search for minutiae is performed by shifting a window trough the binary ridge image, and calculating the number of black and white pixels, n_b and n_w respectively.


bifurcation and the shifting window

If the current pixel is white (valley), and $k = \frac{n_b}{n_w}$ is over a certain rate and the ridge pixels form a contiguous area, the current pixel is a bifurcation. As we can see, in case of a bifurcation, there are more pixels, corresponding to the previously mentioned conditions, so we choose only one of them. The termination is the complementary of the bifurcation, therefore we are able to use the same technique to determine it.

After feature extraction, we get a set of points. But there are three major problems. The application has to be rotation and translation invariant, and has to be able to correct the changes caused by the elasticity of the skin. Most of the existing applications use the global features to estimate the rotation and translation parameters, and correct the initial set of minutiae, according to the estimated values. This method in some cases is inaccurate, therefore corrupts the performance of the application. We did not accept this solution, but developed an error proof method instead. Our method uses a rotation and translation invariant representation, using symbol sets obtained from the distances between minutia, and the angles between the lines connecting them.



symbol set generation from minutiae set

The S symbol set is built as follows:

$$S = S \cup \left\{ s \middle| s = (a_{irj}, \max(d_{ir}, d_{jr}), \min(d_{ir}, d_{jr})), j \in A, i \neq j, r \neq j, d_{ir} \geq d_{jr} \right\}$$

Where A is the set of minutiae, r is the reference minutiae.

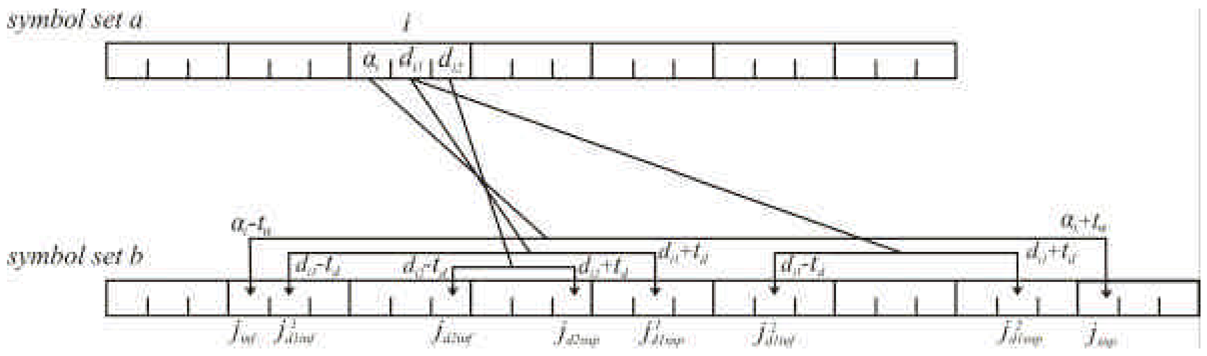
The two reference minutiae from the two different symbol sets of the minutiae sets to be compared, has to match. We cannot determine such reference, so we assume every minutiae to be reference, and so we obtain n symbol sets, where n is the number of minutiae. The symbol sets are ordered increasingly by the angle and distance parameters to ease the match.

The third problem, the deformations caused by the elasticity of skin, is handled in the matching algorithm. The constants are converted into tolerance intervals, and so we allow a certain deviation from the original location.

Let t_J be the threshold value for the angle and t_d for the distance. The condition of match for two symbols, $a = \langle J_a, d_{a1}, d_{a2} \rangle, b = \langle J_b, d_{b1}, d_{b2} \rangle$ is:

$$a(\alpha_a, d_{a1}, d_{a2}) = b(\alpha_b, d_{b1}, d_{b2}) \Leftrightarrow (\alpha_b - t_a) \le \alpha_a \le (\alpha_b - t_a),$$
$$(d_{b1} - t_d) \le d_{a1} \le (d_{b1} - t_d), (d_{b2} - t_d) \le d_{a2} \le (d_{b2} - t_d)$$

Matching two symbol sets, means calculating a matching score, based on the number of matching symbols. To decrease the computational cost of the search, we proposed a fast matching algorithm. For every *i* symbol of the *a* set, we determine the confidence interval of the angle parameter. As we know, the symbol sets are ordered, so there exists the $j_{\inf}, j_{\sup}$ indexes, which represent a fragment of *b*, where every symbol's angle parameter is in the confidence interval of J_i. In this fragment, there are sequences, where the value of J does not change. These sequences are ordered by the second parameter, d_1. Just like in the case of the angle parameter, the confidence intervals of the second parameter are determined. When we scan the symbol sets, the indexes need to be changed only if one of the parameters of *i* had changed. With this algorithm, the first symbol set is scanned only once, in the second symbol set there are only forward steps.



The matching of two fingerprints is performed through comparing every possible combination of the symbol sets of the two minutiae set. A matching score is calculated (from 0 to 100).

References

1. Lin Hong ,Anil Jain. Fingerprint image enhancement: algorithm and performance evaluation. IEEE Transaction on Pattern Analysis and Machine Intelligence 20, 1998.

2. Dario Maio, Davide Maltoni. Direct gray-scale minutiae detection in fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 1997.

3. Anil Jain, Sharath Pankanti: Fingerprint Classification and Matching. MSU-CPS 1999

4. Paul Collier, Fernando Podio. Common Biometric Exchange File Format. NIST/ITL 1999.

5. Kovács-Vajna Miklós Zsolt. A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 2000.

6. Shlomo Greenberg, Mayer Aladjem, Daniel Kogan. Fingerprint Image Enhancement using Filtering Techniques. Real-Time Imaging 8, 2002.

7. Lin Hong, Anil Jain, Sharathcha Pankanti, Ruud Bolle. Fingerprint Enhancement. IEEE WACV, Sarasota, Florida, 1996

8. Jianwei Yang, Lifeng Liu, Tianzi Jiang, Yong Fan. A modified Gaborfilter design method for fingerprint image enhancement. Elsevier Science 2003.

9. D. Dumitrescu, Hariton Costin. Retele Neuronale. Teora, 1996

10. Simon Haykin. Neural Networks. Prentice Hall, 1999.

11. Sergiu Nedevschi. Prelucrarea imaginilor si recunoasterea formelor. Grupul Microinformatica, 1998.