HAND GEOMETRY: A NEW APPROACH FOR FEATURE EXTRACTION

BOREKI, Guilherme[1]; **ZIMMER,** Alessandro[2]
UNICENP – Centro Universitário Positivo, Curitiba - Brazil
Computer Engineering Department
guilherme@boreki.eng.br[1], zimmer@unicenp.br[2]

ABSTRACT

This work presents a complete control access system based on the hand geometry, a hardware key and a vital sign detector. The circuitry reads the hardware key and the heartbeat in order to confirm the identity of the author given by the analysis of the hand image. The presented system distinguish itself from other similar biometric systems mainly because of the feature extraction process which is based on the analysis of the curvature profile of the image, making the system invariant to the rotation and translation of the hand. This makes unnecessary the use of any kind of restriction devices such as pins or pegs to position the hand. FAR rates as low as 0.8% were obtained by the use of simple weighted geometric features on a database of more than 360 hand images.

Introduction

Nowadays it is easy to find biometric devices providing physical access to places or logical access to computer data in several places from large companies to small gyms.
Hand geometry is a kind of biometric measure that is not as diffused in the market as others, nevertheless in 2004 it took 11% of the entire market share for biometric technologies (according to the International Biometric Group). Differently from most of the other systems [1], this work provides a new approach to the way hand geometry features are extracted. Data is read and processed independently of the position of the user hand. This is done by analyzing the curvature profile of the hand contour, making the feature extraction process rotation and translation invariant.
Hand Geometry is a biometric key with medium level of individualization. There are several features that can be extracted and used as key such as finger width and length, overall size of the hand, hand contour among others (a deep study about hand geometry features can be seen in [10]). In this work we obtained satisfactory results by the use of only the weighted mean width and length of the fingers; where the weights were determined by previously analyzing some of the images of our database. Data acquisition was performed by a simple standard desktop-scanner, reading images from the bottom part of the hand. The resulting database is comprised of 80 users with at least 4 images of each user's right hand.
Most of the research done on hand geometry verification use pins between fingers in order to determine specific segmentation areas as can easily be seen in [1] and [2]. In this work the fingers were identified and segmented by analyzing the closed shape formed by the contour of the hand, looking for a sequence of maxima of curvature along the way Our main goal in this project was to be able to acquire the images free from any restriction by allowing the user to put his hand in virtually any position inside the scanning area of the input device. By providing this degree of freedom we are able to achieve a smaller rate of rejection by the users of the system.
The biometric system is completed by two other hardware modules: A hardware key reader and a heartbeat detector. The hardware key provides a unique ID for its owner, and it is used to identify the user to the system in order to retrieve its biometric code stored in the database and compare it to the code extracted from the hand image. The heartbeat detector is used to ensure that the hand placed over the scanner is not a forged image but a real hand. The circuitry attached counts the heart beats and compares it against a threshold to inform the computer about the status of the image acquired. In this paper we focus on our biometric verification methodology and the segmentation and feature extraction processes involved. Figure 1 shows the system's modules.

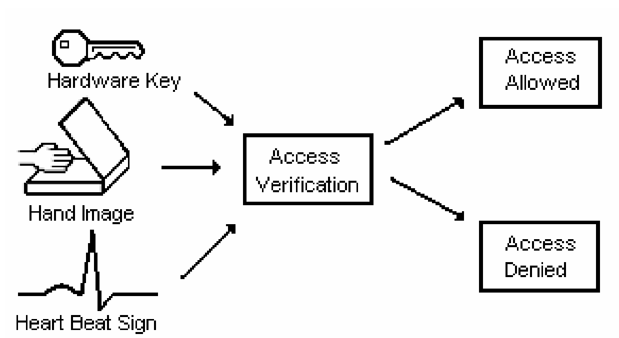

Figure 1 – Systemic view

Into Biometrics

Our feature extraction procedure is divided in two different steps: The contour extraction and the curvature extraction algorithm. The contour is obtained trough binarization and frequency analysis and the curvature is calculated by the use of the DOS (Difference Of Slopes) algorithm [7].

Preprocessing – Contour extraction

With this step we intend to derive a single pixel wide contour of the hand. In Figure 2 we can see one of our scanned hand image sided with the contour we want to extract.

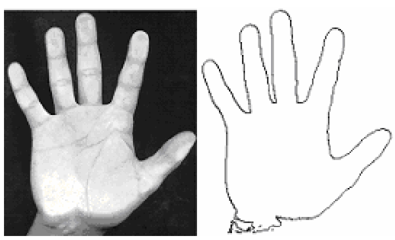

Figure 2 – Original Image and the desired contour

The first thing to be done in order to extract the contour is the binarization – conversion to black & white – of the hand image. This is done by a thresholding method responsible for counting the frequency of pixels of each color (or gray scale in this case) and choosing which one belongs to the background and to the hand. We have opted to use the Otsu auto-thresholding method [3]. Otsu's method uses statistical calculations on the image histogram trying to find an ideal cut-point between the object and the background.
In order to facilitate the threshold determination we have ensured the contrast between the background and the hand by building a black box. This isolation box is put on top of the desktop scanner during the readings. Besides eliminating the external light, it reflects the scanner's own light to the outside of the "reading area" (the light that isn't being read at the moment is reflected outside the scanning trail). Figure 3 shows this box.

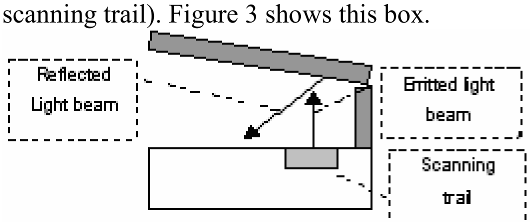

Figure 3 – Isolation Box

To get to the contour of the hand the edges are detected through a frequency analysis of the neighborhood pixels by keeping the areas of high frequency. Figure 4 shows an example of a hand image on which we can carry this kind of analysis.

Figure 4 – Black & white image

The analysis of frequencies is done both in the vertical and horizontal directions ensuring that the width of the contour obtained is one pixel in all its extension. This way we get an image similar to the shown in Figure 2.
To ensure a uniform border, part of the wrist area is automatically cut off and is substituted for a curve with constant curvature in order not to influence the techniques used for biometric features extraction.

Feature Extraction

To reach the desired objective of invariability in rotation and translation of the hand we will use the curvature profile of the contour as the basis for segmenting the fingers from the hand image.
The slope difference is extremely useful when segmenting an image [4],[5]. In this work, the points of high curvature will be used as the segmenting points from which the biometric features of the individual will be taken. These points represent a change of direction of the contour, thus marking the finger's locations. Using such method it is possible to analyze the hand profile without rotating it to a default axis (as seen in other peg-free publications [9]).
As mentioned before, the greatest curvature points where calculated using the DOS (difference of Slopes) technique [7]. This is a method which consists basically in roaming the hand contour with two vectors of the same size and calculating the angle formed by these two vectors. Figure 5 shows how this occurs in a curve. Two vectors V1 and V2 with the same size (in pixels of the contour) must be in sequence. V1' represents the propagation of V1 so an angle will be formed by V2 and V1'. The difference in the curvature is the angle θ formed by the vectors for each pixels of the contour e.g.: Put V1 on the first pixel and calculate θ; move V1 one pixel (also move V2) and find the new θ. Once all θ were found we have a representation of the entire hand curvature which we call the hand curvature profile.
Figure 6 shows the geometric representation of the vectors in a Cartesian chart with the initial position of the

vectors in the origin of the axis. The values of α, β and θ can be calculated applying simple trigonometry relations.
Alpha is the angle between V1’ and X-axis. Beta the angle between V2 and the X axis and Theta, the wanted angle, is formed by V1’ and V2.
Using the relation of the triangle formed by the origin and V1’ we are able to calculate Alpha. Working with a right triangle we can use the relation formed by the hypotenuse and the leg as follows – equation (1).

$$\alpha = \mathrm{Arctg}\left(\frac{\text{Opposite Leg}}{\text{Adjacent Leg}} \right) \tag{1}$$

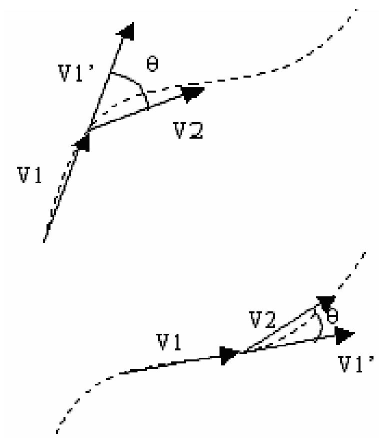

Figure 5 – DOS application in a single curve

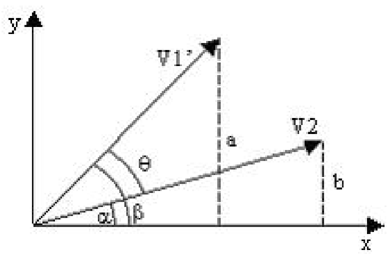

Figure 6 – Angle calculation

The opposite leg it is the height of V1’. The same relation can be applied to V2. Now we have the value of β. θ can easily be calculated as it is the difference between β and α.
As said before, these calculations must be made for all the pixels that comprise the contour of the hand, resulting in a sequence of values that can be analyzed.
This method of extracting the curvature has some particularities [6]. One of these is the change in the quadrant given by the function arctangent. It may cause variations in the curvature graph and can be a problem in future analysis. For these cases an exception treatment is made, analyzing which is the direction of each vector and if necessary changing its signal.

The DOS method works with vectors of defined constant sizes (ω) and may have a group of pixels that can be “jumped” between the vectors (called DOS+ method [7]). The size of the vectors and the space between them are heuristically determined and should be tested and adapted to each different problem. Sometimes DOS+ works a little better than DOS, as in our curvature calculation problem. By the use of a very small space between the vectors (one or two pixels) the results obtained could be improved a little.
The data obtained by applying the difference of slopes on all the contour of the hand will result in a chart like the one shown on Figure 8. Each point represents the angle formed by the vectors in that pixel of the contour.
We can easily notice that the resultant graph has a very interesting characteristic: the points of greatest curvature represent exactly the extremities of the fingers while the portions corresponding to the length of the fingers have a very small curvature variation, almost near to zero.
If we overlap the curvature profile over the hand contour, marking the high curvature points we obtain the segmented regions that can be seen in the Figure 7.

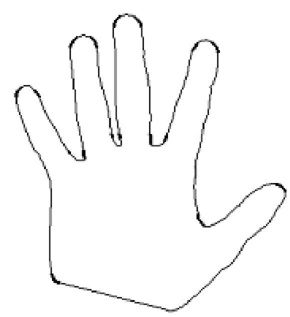

Figure 7 – Contour with the high curvature points marked

Filtering

Tests with the application of the described method showed a variation on the position of the tip of the fingers (the central points between constant curvature regions) to be bigger than 10%. With such variation it becomes impracticable to use it directly to segment the regions since the points obtained are used to estimate the length of the fingers. To solve this problem an additional processing was made as follows.
Analyzing the graph of Figure 8, the marked area that comprises two peaks generally represents either the bending of the tip or the base of a finger. Depending on the curvature value calculated, the tip of the finger can be represented by only one peak of high bending.
Such variations make it difficult to segment the fingers so the signal will have to be filtered or normalized in order to be of any use. To prevent this behavior we will work

with the average point among the high curvature points of the fingertips.

Besides the use of the average points we have applied a Gaussian filter with a kernel twice of the size of ω (size of the vector from DOS method) in order to smooth the curvature profile obtained. Another advantage in using this kind of filter is that, considering that the regions of low curvature are approximately zero, if an isolated peak that does not comprise the region of high curvature appears in the graph it should be attenuated and removed from the analysis.

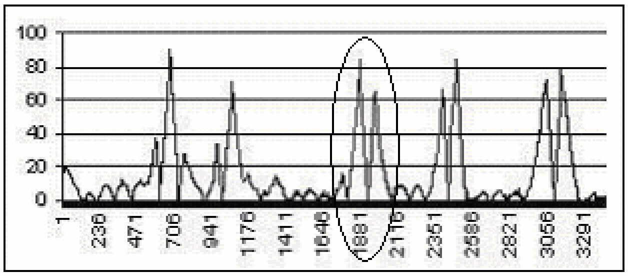

Figure 8 – Curvature of the hand contour extracted by DOS

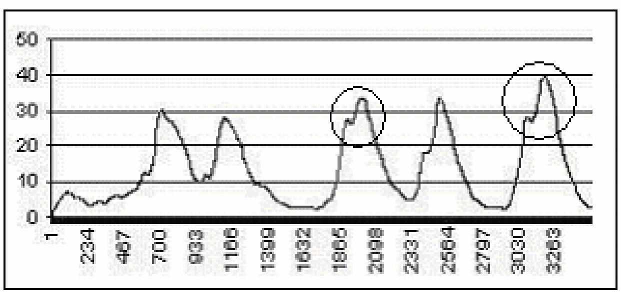

Figure 9 – Gaussian filter applied to DOS

In Figure 9 we see the data of Figure 8 filtered with the Gauss Filter. Even using this kernel size we still have some undesired peaks in the graph. This may cause misinterpretations by the feature extraction algorithm so to solve this we simply applied a common average filter at the output of the Gaussian filter. The new result is show in Figure 10 with a graph that can be successfully analyzed.

In order to find the point in the graph that closely matches the center of a certain fingertip we have not just taken the highest point of a peak, instead we have taken the center point from the whole high curvature points. The curvature points to be considered where the ones higher than the mean curvature of the whole sample. The experiments have shown that the maxima obtained using this algorithm were a closer match to the fingertips.

The mean curvature is the threshold line shown in Figure 11. In this figure we can see which points of the graph are above the average since they are grayed. Now it's easy to calculate the central points which are the points in the exactly middle of any separated high curvature areas.

To find the equivalent points in the original image we just have to count the exact number of points (pixels) by following the contour of the image.

Two of these points require a special treatment: The first ant the last segmenting points that represents the beginning of the thumb and the end of the little finger. This is needed since they are located in low curvature regions and will not be shown detected in any graph. The positions of these two fingers are then estimated using the length of the internal side of the finger-contour plus 15% (percentage find through optimization tests). Even using such estimation these two fingers will have a larger variation than the rest of the fingers.

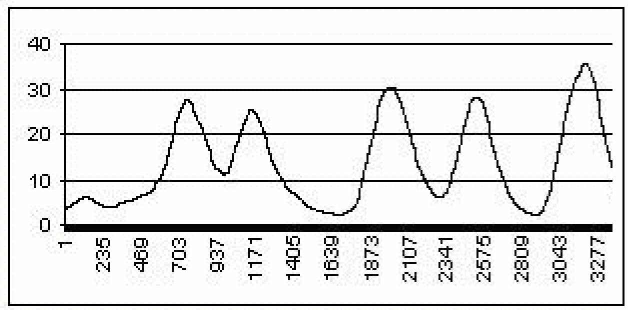

Figure 10 – Average filter applied to Gauss's result

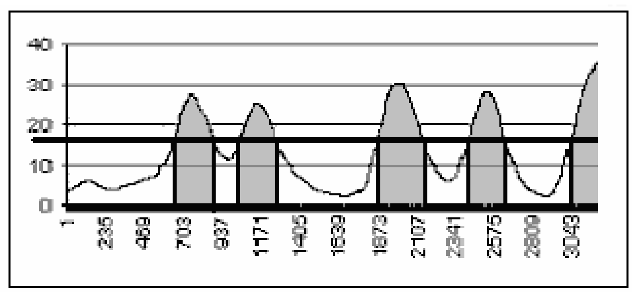

Figure 11 – High Curvature points marked

Feature Measurement

Now we have 11 points. Each of these points represents the central points of the high curvature areas. To find the length of each finger we can draw an imaginary triangle among tree points. The main line of the triangle is then divided into twenty new points. These points will be where the width of the finger will be sampled. A perpendicular line is drawn from the main line of the triangle to the edge of the finger. The average of all the perpendicular lines will result in the finger width.

A picture with these features shown graphically can be seen in Figure 12. The dashed line in the center of each finger represents the length of the finger. The side lines are the lines that will comprise the width of the finger.

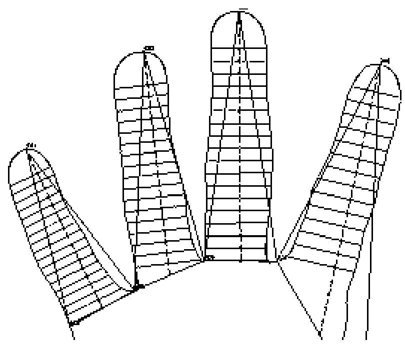

Figure 12 – Picture with length and width

At this point we have five measures of width and five measures of length for each hand. In order to create a biocode for each user, we need to somehow extract a subset of information valuable enough to assure the individualization of the user on the system.

Based on the data samples we collected for each user we have measured the between-class and within-class variation of length and width of each user's fingers in order to find an equation able to account for such variations. We have found two single weighted measures representing the average length and width of the fingers in one hand image and used them as the individual biocode for this user. The resultant equations are as follows:

$$L = 0\ L_{F1} + 2L_{F2} + 2L_{F3} + L_{F4} + 0.2\ L_{F5} \quad (2)$$

$$W = 0\ W_{F1} + 2W_{F2} + 2W_{F3} + W_{F4} + 0.2\ W_{F5} \quad (3)$$

Legend:

- L is the mean value for the finger's length;
- W is the mean value for the finger's width;
- L_{F1}, L_{F2}, ..., L_{F5} are the length of each finger starting from the thumb.
- W_{F1}, W_{F2}, ..., W_{F5} are the width of each finger starting from the thumb.

In both cases (Length and Width) the thumb is rejected since its features are too unstable [8].

As can be seen from the equations, the best results were obtained by completely eliminating the thumb influence since the measurements obtained presented a great deal of variation and were consequently unreliable.

Classification

The classification's job is to confirm or deny the claimed identity of the user based on the biocode extracted from the hand image. Our distance-based classifier used the variation of the length and width of Equations 2 end 3 in order to accept or reject the user.

We have stipulated that for a candidate to be accepted it should have the calculated length and width inside a region that comprises two times the standard deviation of the training user's samples. This criterion resulted in an elliptical acceptance area.

Figure 13 shows the FAR x FRR graph, where the threshold value is the multiplier of the standard deviation. With this graph we are able to see that the Equal Error Rate (ERR) is about 4.5%.

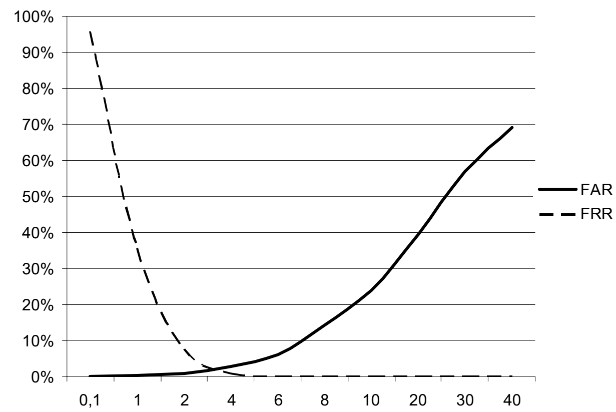

Figure 13 – FAR X FRR Graph

In order to calculate the standard biometric code for a given person we have used four different hand images. The image having the most distant biocode value from the average was discarded and the average was then recalculated.

Results

We created a database with the right hand images of 80 people (male and female - 17 to 38 years old), with at least four readings of each hand. The voluntaries were asked to change the position of their hand and the degree they opened their fingers in order to generate completely different images. Next, the feature extraction and standard biocode calculation was done as described before. Figures 14 to 17 show several sample images processed by the proposed algorithm. They show clearly the flexibility of the method in relation to the rotation and translation of the hands. The false rejection rate (FRR) was determined based on the average rejection rate for each one of the users considering all the authentic images available for that user. The false acceptance rate (FAR) was

determined by fixing a user and using all the other user's images for testing. The final rate considered was the average FAR rate for all of the 80 users.
A false acceptance rate of only 0.8% and a false rejection rate of 3.8% were obtained with this methodology.

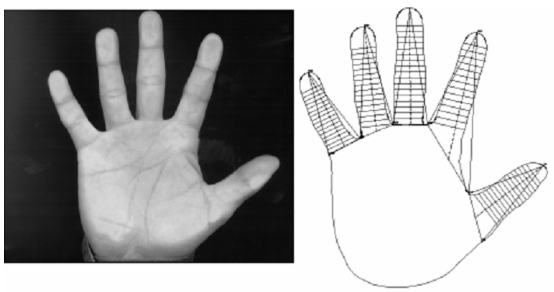

Figure 14 – Usual hand position

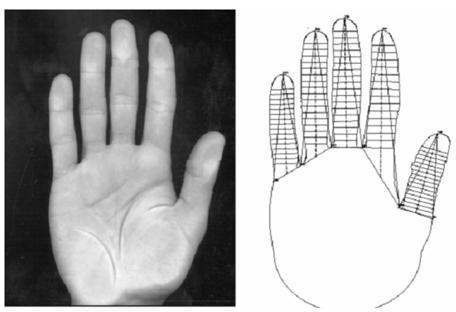

Figure 15 – Nearly-closed-fingers hand

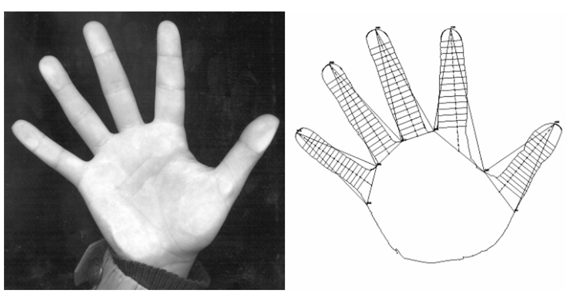

Figure 16 – Twisted hand

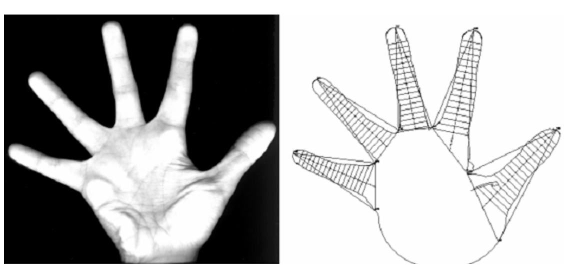

Figure 17 – Wide-open hand

Conclusion

In this work we developed a new approach to hand geometry feature extraction by using curvature analysis. This way we can extract features without imposing any restriction to the user which makes it possible to identify the hand with virtually any rotation and translation.
The results were impressive since we got an FAR of 0.8% and a FRR of 3.8%.
This project is part of a control access system that uses a hardware key for the user's identification and confirms the readings from the scanner against a vital sign detector. The system is currently being improved by the addition of the analysis of the palm print and by the use of a different classifier.

Reference

[1] R. Sanchez-Reillo, C. Sanchez-Avila, A. Gonzalez-Marcos, "Biometric Identification through Hand Geometry Measurements", IEEE Trans. On Pattern Analysis and Machine Intelligence v22 n10, pp. 1168-1171, 2000.

[2] A. Jain, Arun Ross, "A Prototype Hand Geometry-based Verification System". Proc. of 2nd Inter. Conf. on Audio- and Video-based Biom. Person Auth. AVBPA, pp.166-171, 1999.

[3] Otsu N., "A Threshold Selection Method from Gray-level Histograms", IEEE Trans. Syst. Man, Cybern., vol. 9, no. 1, pp. 377-393, 1979.

[4] M. Worring, A. W. M. Smeulders, "The Accuracy and Precision of Curvature Estimation Methods" , Proceedings of the ICPR: Image, Speech and Signal, Analysis, The Hague, 1992.

[5] L. O' Gorman, "An Analysis of Feature Detectability from Curvature Estimation", IEEE Conf. Comput. Vision and Pattern Recognit, pp. 235-240, 1988.

[6] A. Zimmer, L.L. Ling, "Preprocessing: Segmenting by Stroke Complexity", Proceedings of the VI Iber-American Symposium on Pattern Recognition, pp.89-94, 2001.

[7] L. O'Gorman, "Curvilinear Feature Detection from Curvature Estimation", Proceedings of the 9th International Conference on Pattern Recognition, pp. 1116-1119, 1988

[8] Oden, A. Ercil, and B. Buke, "Combining implicit polynomials and geometric features for hand recognition," Pattern Recognition Letters, vol. 24, pp. 2145-2152, 2003.

[9] Alexandra L.N. Wong and Pengcheng Shi, "Peg-Free Hand Geometry Recognition Using Hierarchical Geometry and Shape Matching", IAPR Workshop on Machine Vision Applications (MVA02), 2002.

[10] S; Travieso C.M.; Alonso, J.B.; Ferrer M.A.; "Automatic biometric identification system by hand geometry", IEEE 37th Annual 2003 International Carnahan Conference on 14-16, 281 – 284, Oct. 2003.