

# Dynamic threshold using polynomial surface regression with application to the binarisation of fingerprints

Krzysztof Mieloch<sup>a</sup>, Preda Mihailescu<sup>b</sup> and Axel Munk<sup>c</sup>

<sup>a</sup>University of Göttingen, Göttingen, Germany;

<sup>b</sup>University of Paderborn, Paderborn, Germany;

<sup>c</sup>University of Göttingen, Göttingen, Germany

## ABSTRACT

Dynamic thresholds are a popular and effective means of binarisation, used in image processing. We show that the method is a very special case of a general setting in which input data from a background window are fitted with a polynomial surface after which the data from a smaller, embedded focus window are thresholded with respect to the fitted surface.

The method has been implemented and used with good results in the context of fingerprint recognition.

**Keywords:** binarisation, fingerprint, thresholding, polynomial fitting

## 1. INTRODUCTION

A common model for edge enhancing and thresholding in the context of fingerprint recognition<sup>1,2</sup> is inspired by the study of human visual perception. It consists in gathering context information about contrast and gray value distribution from a medium sized *background* window which contains the *focus window* which will actually be processed. The data in the focus window are thresholded by using the background window information.

The dynamic thresholding method is used for binarising gray level images. The procedure consists in computing the average gray value of a square of fixed size surrounding the pixel for each pixel of an image. The pixel value is then set to black or white, according to whether its initial value was larger than the surrounding average or not.

This method may be considered as a two window model in which the inner focus window is reduced to a single point, which, typically, will be the mean of a given set of values and hence a least squares fit by a constant. One may thus consider the dynamic thresholding as a method in which the data form a *background window*. These data are fitted by least squares with a constant surface. Within the *focus window* the gray scale data are binarised by comparing to the fitting values for the background window.

This analysis of the usual threshold binarisation suggests a natural generalisation:

- (i) Replace the fitting constant by a general fitting polynomial surface.
- (ii) Let the size of both background and focus window vary.
- (iii) Use a sampling step  $s$  – thus allowing  $s > 1$  – within the background window, for computing the fitting surface. This is done for computation economy.

---

Further author information: (Send correspondence to Krzysztof Mieloch)

Krzysztof Mieloch: E-mail: mieloch@math.uni-goettingen.de, Telephone: +49 551 39 13514

Preda Mihailescu: E-mail: preda@upb.de, Telephone: +49 5251 60

Axel Munk: E-mail: munk@math.uni-goettingen.de, Telephone: +49 551 39 13501

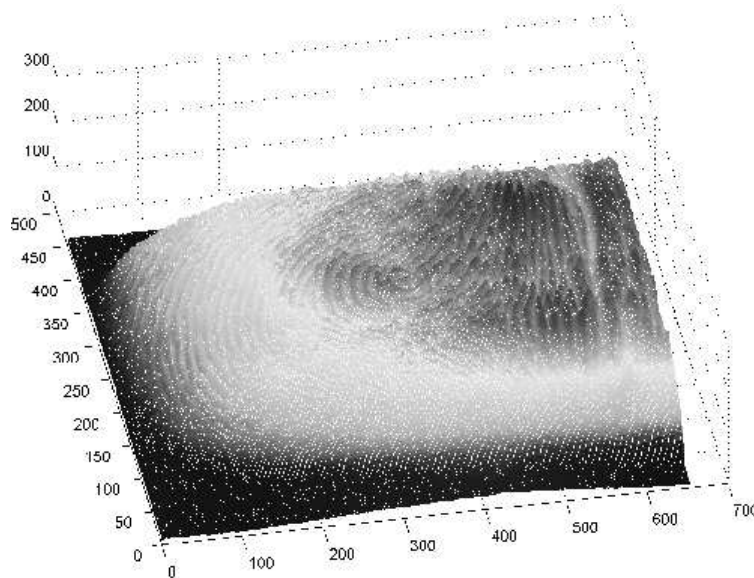
In the classical setting described above dynamic threshold binarisation is used in various applications of image processing.<sup>3</sup> It is thus likely that the generalisation we present here may be adapted to a wide range of image processing tasks. We developed and optimised it in the context of fingerprint recognition and we will restrict our presentation to this case. Therefore, the bulk of this paper is dedicated in detail to this application. For this context we give detailed formulae, pseudocode and visual results of the method. In particular, we show how careful data management and optimised algorithms can lead to the apparently paradoxical consequence that surface fitting can be done more efficiently than average value fitting in the classical way. The results of this part of the paper can thus be used *as is* by fingerprinting specialists.

## 2. BINARISATION FOR FINGERPRINT RECOGNITION

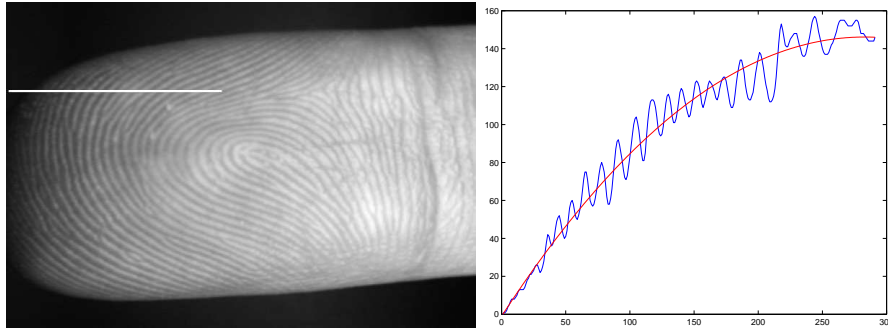
Images of fingerprints used for machine - recognition may come from various sources, such as visual scanners, thermo scanners, touchless scanners or even scanned images of ink imprints gathered on paper. In case of some visual and touchless scanners which have recently received much attention because of their advantages, the three dimensional gray scale fingerprint (with the  $z$  - axis, or height, plotting the gray scale value, usually in the range  $0 - 255$ ) appears like a wavelike perturbation of a more or less continuous surface. An example of a three dimensional plot of a fingerprint is presented in Figure 1. In Figure 2 we have displayed an image of a cross section of this plot. The perturbations are the fingerprint ridges: the main topic of interest for recognition. The background surface is the visual image of the finger shape itself. Experience shows that physical factors such as blood pressure can influence the smoothness of this surface: scans of fingers with low blood pressure tend to display distortions of the finger surface which do not exist in reality and are generated by variations of heat and reflection on the finger surface. Since the sole interest of fingerprint recognition is focused on ridges, it is clear that an accurate isolation of the ridges from the background surface is of great importance; moreover, given the natural curvatures of this surface, flat fitting may lack accuracy.

The problem of accurately determining line ridges by binarisation of visual scans, together with the theoretical observation on the classical dynamic threshold method made in the introduction, lead to our investigations.

Some remarks concerning performance are important in our context. Binarisation is a *preprocessing* step of the *minutiae extraction* in fingerprint recognition. Although the ulterior steps of the process highly depend on the accuracy of binarisations, it is requested that preprocessing would only take a small amount of the time used for the total data extraction process. Most published techniques do use binarisation. However, Maio et. al.



**Figure 1.** Three dimensional picture of fingerprint.



**Figure 2.** The right plot shows the gray scale values of the fingerprint image along the straight line shown on the left picture

proposed in<sup>9,10</sup> an approach for direct gray scale data extraction. This indicates that reducing the binarisation time may be considered as so important as to even try to eliminate it altogether.

Under these circumstances the performance goals we set for our new binarisation technique are the following:

1. Clearly improve the accuracy of binarisation with respect to the current dynamic threshold method, by explicitly taking the geometric characteristics of the background finger shape.
2. Optimise the resulting regression computation over the whole finger surface, so that in the worst case no loss of performance with respect to the simpler constant fitting be registered.
3. Comparison of our method to approaches which avoid binarisation, but certainly need on-line surrogates for compensating for the loss of this preprocessing. The time/accuracy account should be favourable in our approach.

The method we present here satisfies these criteria and has been intensively tested on various finger scans. It behaves satisfactorily even for imprints which have an essentially flat background (see Figures 6 and 7) and are thus not the primary targets for our approach. This is particularly practical since it allows to use a global binarisation method rather than having to make choices in dependence on the scan type. The robustness of our method may rely upon its good data management and numeric adaptability. It should be mentioned that further robustification of the method presented can be achieved, of course, by the use of different loss functions, such as  $L^1$ -loss or others. This will improve the performance, however, the computational effort will increase.

As has been mentioned, the method relies on fitting polynomial surfaces to data from background windows of gray scale values. Spline surfaces<sup>4</sup> are an alternative which we have also tested without satisfactory gain in accuracy/performance.

## 2.1. Other binarisation methods

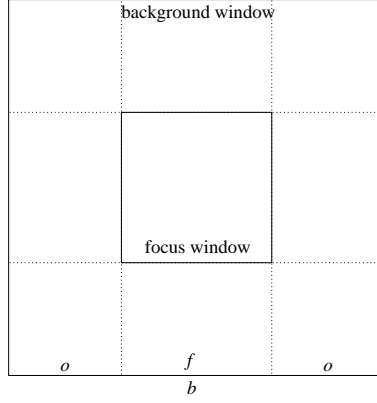
1. *Global thresholding.* Here, a global threshold-parameter  $T$  is to be found, and then binarisation is proceeded as follows:

$$B(x, y) = \begin{cases} 1 & \text{if } G(x, y) \geq T, \\ 0 & \text{otherwise.} \end{cases}$$

Finding of threshold:

- A simple option to find  $T$  is to choose the mean of the pixels  $\mu = \frac{\sum \sum G(i, j)}{n \cdot m}$
- Histogram analysis. Threshold is chosen, so that it falls between high peaks of histogram

The threshold is computed either global for the whole picture, or the image is divided into small regions, for each the threshold is determined. Because of variations in fingerprint image (e.g. produced by unequal pressure of the finger at the scanner) it is often not feasible to use an optimal global threshold.



**Figure 3.** Background and focus windows.

2. *Using of the directional field.* Here these points indicating ridges one selects the local maximum gray-level values along a direction normal to the local ridge direction.<sup>12</sup>
3. *Using of edge-detector.* With use of edge filters, the edges of ridges are found.

### 3. THE REGRESSION EQUATIONS

Let  $\Gamma = \{G(x, y) \in [0, 255] : 0 \leq x < W, 0 \leq y < H\}$  be a gray level image. If  $n$  is an odd integer,  $N_n[x, y]$  is the neighbourhood of the point  $x, y$  consisting of a square of size  $n$ , symmetric around  $(x, y)$ , i.e.

$$N_n[x, y] = \{(u, v) \in \Gamma : |u - x| \leq \frac{n-1}{2} \text{ and } |v - y| \leq \frac{n-1}{2}\}. \quad (1)$$

If  $(a, b)$  are the coordinates of a point with respect to the window  $N_n[x, y]$ , its absolute coordinates will thus be  $(x + a, y + b)$ , or, allowing point addition,  $(x, y) + (a, b)$ . Note that here we choose the origin of the window in its centre, thus obtaining negative local coordinates. If one wishes to work in positive integers, one can, for example, situate the origin at the lower left corner of  $N_n[x, y]$  and add the vector of the centre point as a correction to the previous formula.

If  $\mu_n(x, y) = \sum_{(u,v) \in N_n(x,y)} G(u, v)/n^2$  is the arithmetic mean of the pixel values in the neighbourhood  $N_n(x, y)$ , dynamic thresholding consists in replacing  $G(x, y)$  by the binary value

$$B(x, y) = \begin{cases} 1 & \text{if } G(x, y) \geq \mu_n(x, y), \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that hereby the mean  $\mu_n(x, y)$  is to be estimated essentially for every single point  $(x, y)$ . Indeed, the low accuracy of the mean value fitting calls for more frequent updates of the threshold  $\mu(x, y)$ .

Let  $b, o$  and  $f$  be positive integers with  $b = f + 2o$ : the sizes of a background and a focus window (see Figure 3);  $o$  will be the overlapping width of the background windows while the focus windows of size  $f$  slide horizontally and then vertically over the given image  $\Gamma$  (see Figure 4). Let  $d$  be a given degree; the general bivariate polynomial of degree  $d$  may be written as:

$$p_d(x, y) = \sum_{i+j=0}^d a_{i,j} \cdot x^i y^j; \quad (3)$$

note that the sum of the powers of  $x$  and  $y$  (the weight of  $p_d(x, y)$ ) is bounded by  $d$ .

Let  $\beta_b, \varphi_f \subset \Gamma$  be windows in  $\Gamma$  of sizes  $b$  respectively  $f$  as above, such that  $\beta$  is symmetrically containing  $\varphi$ . If  $b, f$  are odd, one may think of  $\beta$  and  $\varphi$  as two different sized neighbourhoods of a given point situated at their common centre, in the sense of 1.

We will deduce the equations for the coefficients of a polynomial  $p_d$  which best fits the data in  $\beta$ , in the sense of least squares. Let the data be sampled in steps of length  $s \geq 1$ : a step  $s > 1$  will produce a reduction of the computation and may be useful for larger sized windows. In that case the conditions  $s|b$  and  $s|f$  are requested. Let

$$\{ \beta(x, y) : 0 \leq x, y < b \}$$

be the pixel values of the background window  $\beta$ . By the symmetry condition, the focus window is

$$\varphi = N_f(x, y).$$

The quadratic error produced by replacing the gray values in  $\beta$  by the values of a generic bivariate polynomial  $p_d$  of degree  $d$  is:

$$\epsilon(p_d) = \sum_{x,y=0}^{b/s-1} ( p_d(sx, sy) - \beta(sx, sy) )^2. \quad (4)$$

The coefficients of the least square approximant  $p_d$  which minimises the error  $\epsilon(p_d)$  are obtained by taking partial derivatives in 4 with respect to the coefficients  $a_{i,j}$  of  $p_d$  in 3 and setting them to 0. After some computations, this leads to the normal equations of regression:

$$\sum_{i+j=0}^d a_{i,j} \left( \sum_{x,y=0}^{b/s-1} (sx)^{i+k} \cdot (sy)^{j+l} \right) = \sum_{x,y=0}^{b/s-1} \beta(sx, sy) \cdot (sx)^k \cdot (sy)^l, \quad (5)$$

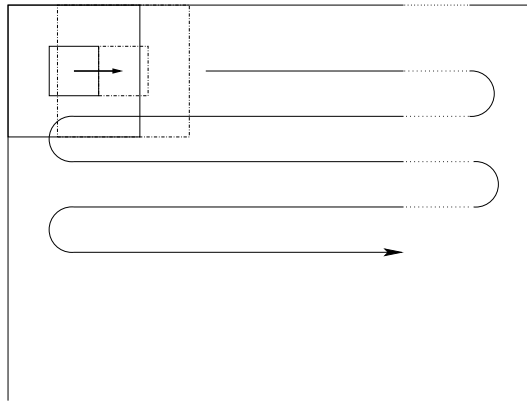
for  $k + l = 0, 1, \dots, d$ .

This yields a linear system of  $D = (d + 1)(d + 2)/2$  equations in the unknown coefficients  $a_{i,j}$  of  $p_d$ . In order for the system to have a solution,  $s$  must be chosen such that

$$(b/s)^2 \geq D \quad (6)$$

For the ease of the discussion, let

$$\begin{aligned} \mu(k, l) &= \sum_{x,y=0}^{b/s-1} \beta(sx, sy) \cdot (sx)^k \cdot (sy)^l \quad \text{and} \\ \sigma(m, n) &= \sum_{x,y=0}^{b/s-1} (sx)^m \cdot (sy)^n \\ &= \sum_{x=0}^{b/s-1} (sx)^m \cdot \sum_{y=0}^{b/s-1} (sy)^n. \end{aligned} \quad (7)$$



**Figure 4.** The sliding of the windows over the image.

Let us also define the following operations on ordered pairs of integers:

$$\begin{aligned}(i, j) + (k, l) &:= (i + j, k + l) \quad \text{and} \\ \iota^{-1}(m, n) &:= \frac{(m + n)(m + n + 1)}{2} + m.\end{aligned}$$

Note that hereby,  $\iota^{-1} : \mathbb{N}^2 \rightarrow \mathbb{N}$  is a bijective counting of pairs of integers; it is thus indeed the inverse of a bijective map  $\iota : \mathbb{N} \rightarrow \mathbb{N}^2$ . We shall use this map for enumerating the pairs  $(i, j) : i + j \leq d$  in terms of a single integer  $n \leq D$ : thus we have

$$\forall n \leq D \quad : \quad \iota(n) = (i, j) \quad \text{with} \quad i + j \leq d.$$

This notation simply brings in evidence the fact that our linear system with double indexed unknowns  $a_{i,j}$  is in fact a simple linear system in  $D$  unknowns, which is transformed into a system with double indexed unknowns by using  $\iota$ . With these prerequisites, the system 5 may be written as

$$\sum_{i+j=0}^d a_{i,j} \cdot \sigma(i + k, j + l) = \mu(k, l), \quad \text{for } k + l = 0, 1, \dots, d.$$

This can be rewritten in terms of matrices as

$$\mathbf{A}(d) \cdot \mathbf{a} = \mathbf{M}, \tag{8}$$

with

$$\begin{aligned}\mathbf{A}(d)_{m,n} &= \sigma(\iota(m) + \iota(n)), \\ \mathbf{a}_n &= a_{\iota(n)}, \quad \text{and} \\ \mathbf{M}_n &= \mu(\iota(n)), \quad \text{for } m, n = 0, 1, \dots, D - 1.\end{aligned} \tag{9}$$

The dynamical threshold method with polynomial regression consists in the following:

- I. Fix the parameters  $b, f, s$  and  $d$ . Note that  $o = b - 2f$  is herewith fixed, too.
- II. Position the background window: start in a corner of the image and slide this background in horizontal and then vertical directions along the image.
- III. For each position of the background window  $\beta(x, y)$ , compute the corresponding regression polynomial  $p_d = p_d[x, y]$ .
- IV. For each point  $(a, b)$  in the focus window  $\varphi(x, y)$ , estimate the polynomial  $p_d[x, y]$  at  $(a, b)$  and compare the image data  $G(a + x, b + y)$  with the the predicted values  $p_d[x, y](a, b)$ , like in 2.
- V. Reposition the background window by shifting in steps of  $f$ , horizontally and vertically, until reaching the diametral corner to the starting one. For each new position repeat III. and IV.

#### 4. DATA MANAGEMENT

It is essential to avoid repetition of identical operations in order to reduce the computation cost. We shall concentrate in this section on some procedures for optimising the performance.

First, note that the matrix  $\mathbf{A}(d)$  does not depend upon the image data or the position of the background window 7, 8. In fact, it only depends upon the size parameters  $b$  and  $s$  and the degree  $d$ . For fixed  $b, s$  it also has the nice inheritance property consisting in the fact the  $\mathbf{A}(d') \subset \mathbf{A}(d)$  for  $d' < d$ , in the sense that  $\mathbf{A}(d')$  is the submatrix of size  $(d' + 1)(d' + 2)/2$  of  $\mathbf{A}(d)$  situated in its upper left corner.

Since the system 8 needs to be solved for each position of the background window, factoring the matrix  $\mathbf{A}(d)$  in its  $\mathbf{L} \cdot \mathbf{U}$  decomposition<sup>5</sup> before running the algorithm will reduce the system solving times form  $\mathcal{O}(D^3)$  to  $\mathcal{O}(D^2)$  operations. Since all the entries of  $\mathbf{A}$  are essentially products of sums of different powers of the integers

$< (b/s)$ , the matrix can be factorised in its general shape using symbolic computation. This we have done using the PARI<sup>11</sup> package for degrees  $d < 9$  and it has the effect of allowing integer arithmetic instead of floating point : the common denominators of the divisors involved in the solution of 8 have nice algebraic closed shapes.

The right hand side of 8 carries the image data and global precomputations are thus not possible. It is however possible and important to take overlaps in consideration for minimising the update costs for the data  $\mu(x, y)$ . Since the background window is shifted only by the size  $f$  of the focus window at each step, a certain amount of data will contribute to the values  $\mu(k, l)$  for more than one window. We use a procedure which consists in adding separately the column data and then accumulating them to produce the window data in  $\mu(k, l)$ . Let us thus define

$$Y_l(x, y) = \sum_{i=0}^{b/s-1} (s \cdot i)^l \cdot G(x, y + i \cdot s),$$

and

$$\begin{aligned} \mu_{x,y}(k, l) &= \sum_{i,j=0}^{b/s-1} G(x + is, y + js) \cdot (si)^k \cdot (sj)^l \\ &= \sum_{i=0}^{b/s-1} (si)^k \cdot Y_l(x + is, y). \end{aligned} \quad (10)$$

For fixed  $y$ , the values  $Y_l(x, y)$  will be computed only once and then be used for computing all  $\mu_{x,y}(k, l)$  with that given  $y$ -value. At a horizontal shift, only  $f/s$  new  $Y$ -values will replace the first  $f/s$  values involved in the current  $\mu_{x,y}(k, l)$  sums; similarly, at a vertical shift, only  $f/s$  new  $G(x, y)$  values will replace the first ones in a  $Y$ -sum. We shall split these operation in a sequence of  $f/s$  steps, which consist in replacing a starting value by a new value in a given sum. The following lemma formalises the situation:

LEMMA 4.1. *Let  $V = \{ v(n) \in \mathbb{N} : n = 0, 1, \dots, N \}$  a set of data,  $b, s, k > 0$  integer parameters such that  $s^2 \ll N$  and consider the following evaluations on the set  $V$ :*

$$E_k(u) = \sum_{i=0}^{b/s-1} (is)^k \cdot v(u + is),$$

and the differences

$$\Delta(E_k(u)) = E_k(u + s) - E_k(u).$$

Then the followings recursion holds:

$$(i) \quad \Delta(E_0(u)) = v(u + b) - v(u),$$

$$(ii) \quad \Delta(E_k(u)) = b^k \cdot v(u + b) - \sum_{i=0}^{k-1} \binom{i}{k} s^{k-i} E_i(u + s).$$

*Proof.* Item (i) is a direct computation:  $E(u + s) - E(u) = \sum_{i=1}^{b/s} v(u + is) - \sum_{i=0}^{b/s-1} v(u + is) = v(u + b) - v(u)$ . Item (ii) follows after some careful regrouping of terms in the equality.  $\square$

All the recursive computations needed for the updating of the values  $\mu(x, y)$  when the background window slides along the gray value image are obtained by applying the above lemma for various choices of  $V$ . We first apply the lemma to the vertical shift. In this case  $v(y) = G(x, y)$ , for fixed  $x$  and  $E_l(y) = Y_l(x, y)$ . Lemma 4.1 provides the values of the step shifts when  $y$  is increased by  $s$ .

For the horizontal shift, consider the data  $v(x) = Y_l(x, y)$ , for fixed  $l$  and  $y$ . The lemma provides us the step changes  $\mu_{x+s,y}(k, l) - \mu_{x,y}(k, l)$  for the fixed  $l$ . It will thus have to be applied for  $d$  different values of  $l$ , in order to update all the  $\mu$ -values in the right term of 8. After computing the current fitting polynomial  $p_d[x, y]$ , the binarisation requires its evaluation at all the points within the small focus window. Here again operation can be spared by adequate finite difference operations. We leave it to the reader to develop his preferred algorithm for this small substep.

We have so far gathered the informations for an algorithm for dynamic thresholding with polynomial surface regression. We express this algorithm below in detail, using pseudocode description. The input of the algorithm is the following:

**INPUT:**

1. A gray value image  $\mathbf{G} = \{G(x, y) : 0 \leq x < W; 0 \leq y < H; 0 \leq G(x, y) < 256\}$ .
2. The auxiliary parameters  $b, f$  which are the (odd) sizes of the background and the focus windows, respectively. The overlap is  $o = (b - f)/2$  and at each step the background window is shifted by  $o$  horizontally or vertically.
3. The sampling step  $s \geq 1$  and the degree of the fitting polynomial  $d \geq 0$ . Note that for  $d = 0$  we have the classical dynamic threshold method, with an improved data management.

The output is a binary image  $\mathbf{B} = B(x, y)$  of the same size as the gray scale image.

For the linear algebra steps for L-U decomposition and solution of the linear system 8 any standard solution can be used.

## 5. PRACTICAL EXPERIENCE

The procedure described above has been used on a large scale of fingerprint images obtained by a touchless device with various scan qualities (for example see figure 5). The approach is considered to perform better if for comparable run times (for dynamic moments this implies an adequate choice of the degree and sampling step, such as to keep the pace with the simpler average estimate) the resulting binarisation is more accurate. We have not develop an abstract accuracy metric; rather the accuracy is evaluated both visually and by integration in our own minutiae extraction algorithm. The dynamic threshold method has found to perform overall better, in particular in images of less quality, where it could extract more contrast.

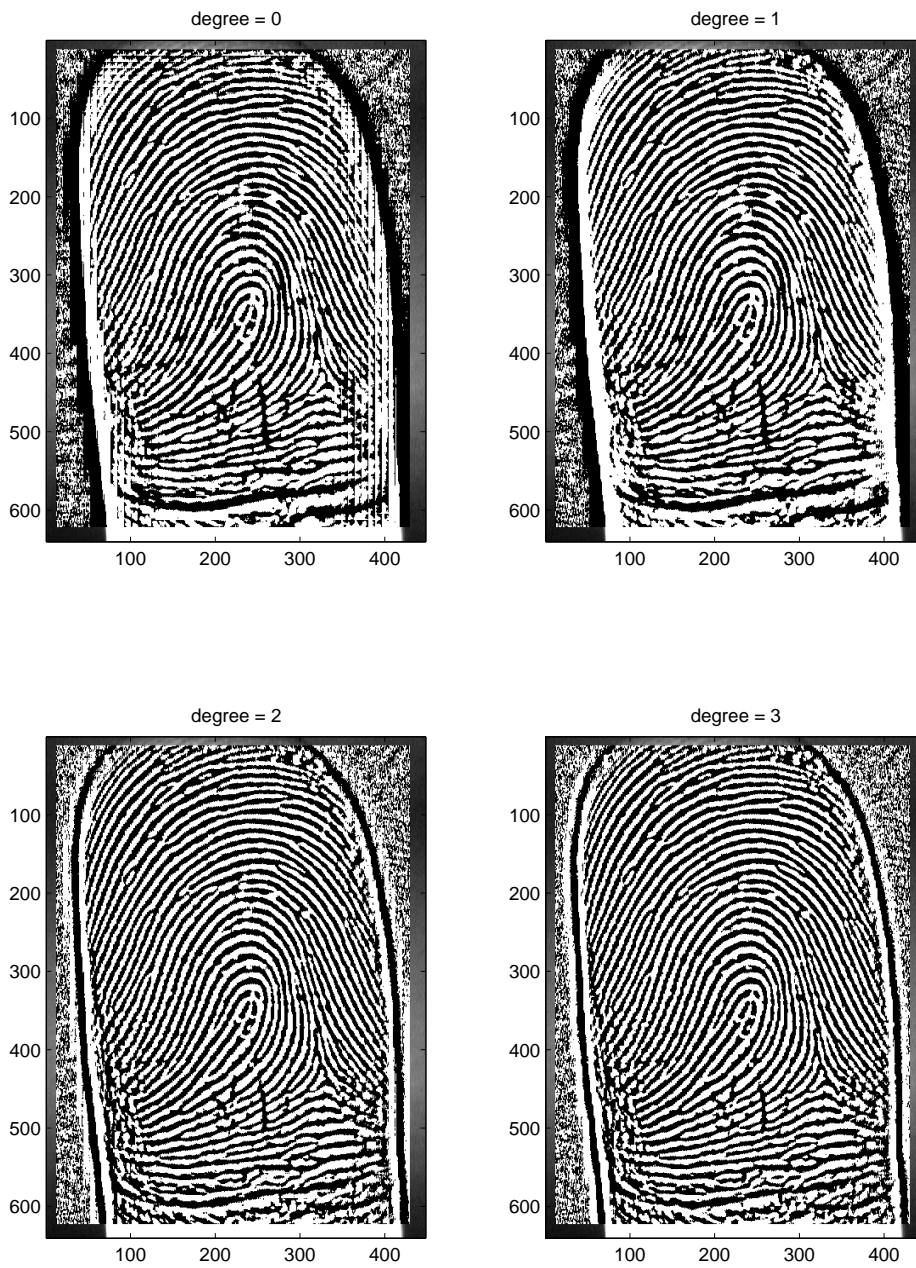
In fig. 5 we can observe that significant improvement of the binarisation quality when the degree changes its value from 0 to 1 and from 1 to 2, whereas there is hardly improvement of quality when increasing the degree of polynomial from 2 to 3. Indeed, in our experiments the degree of 2 has been found to be sufficient.

The tests for "flat" fingerprint images (example - fig. 6) has shown there is no improvement by increasing the degree value (fig. 7). This implies, the polynomial based thresholding works for this type of fingerprint images as good as the standard mean thresholding.

## REFERENCES

1. D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar: *Handbook of Fingerprint Recognition*, Springer Verlag, June 2003.
2. A. K. Jain and S. Pankanti: *"Automated Fingerprint Identification and Imaging Systems"*, Advances in Fingerprint Technology, 2nd Ed. (H. C. Lee and R. E. Gaensslen), CRC Press, 2001.
3. B. Jähne: *"Digital Image Processing"*, Springer Verlag, 2002
4. P. Diercks: *Curve and Surface Fitting with Splines*, Monographs on Numerical Analysis, Oxford University Press (1995)
5. G. H. Golub and Ch. van Loan: *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1983.
6. L. Hong, Y. Wan and A.K. Jain: *"Fingerprint Image Enhancement: Algorithms and Performance Evaluation"*, Proc. IEEE Comp. Soc. Workshop on Empirical Evaluation Techniques in Computer Vision, Santa Barbara. June 1998, pp. 117-134.
7. L. Hong, Y. Wan and A.K. Jain: *"Fingerprint Image Enhancement: Algorithms and Performance Evaluation"*, IEEE Transactions on PAMI, Vol. 20, No. 8, pp.777-789, August 1998.
8. L. Hong, A.K. Jain, S. Pankanti and R. Bolle: *"Fingerprint Enhancement"*, Proc. IEEE Workshop on Applications of Computer Vision, Sarasota, Fl, pg. 202-207, Dec. 1996.





**Figure 5.** Polynomial based thresholding with window size  $32 \times 32$ , size of core  $f = 4$  and step  $s = 2$  of the fingerprint presented in Fig. 2.



**Figure 6.** Example of a flat background fingerprint obtained from a thermoscanner.



**Figure 7.** Polynomial based thresholding with window size  $32 \times 32$ , size of core  $f = 4$  and step  $s = 2$  of the imprint presented in Fig. 6.

9. D. Maio and D. Maltoni: "*Direct Gray-Scale Minutiae Detection in Fingerprints*", IEEE Transactions on Pattern Analysis Machine Intelligence, vol.19, no.1, pp.27-40, 1997.
10. D. Maio and D. Maltoni: "*Minutiae extraction and filtering from gray-scale images*", in L.C. Jain, U. Halici, I. Hayashi, S.B. Lee, Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, 1999.
11. H. Cohen et. num. al.: *PARI/GP, Laboratoire A2X*, <http://pari.math.u-bordeaux.fr>
12. R. Bolle, A. Senior, N. Ratha and S. Pankanti: "*Fingerprint Minutiae: A Constructive Definition*", Lecture Notes in Computer Science, Vol. 2359/2002, p. 58-66