Stanislav Bartoň

Vectorization of bitmaps based on the LSQ method

In: Jan Chleboun and Petr Přikryl and Karel Segeth and Jakub Šístek (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 6-11, 2010. Institute of Mathematics AS CR, Prague, 2010. pp. 9--14.

Persistent URL: http://dml.cz/dmlcz/702734

VECTORIZATION OF BITMAPS BASED ON THE LSQ METHOD*

Stanislav Bartoň

Abstract

The paper presents the software procedure (using MAPLE 13) intended for a considerable reduction of digital image data set to a more easily treatable extent. The photos taken in high resolution (and corresponding data sets) contain coordinates of thousands of pixels, polygons, vertexes. Presented approach substitutes this polygon by the new one, where a smaller number of vertexes is used. The task is solved by means of adapted least squares method. The presented algorithm enables the reduction of number of vertexes to 5% of its original extent with an acceptable accuracy ± one pixel (i.e. distance between the initial and the final polygon). The procedure can be used for processing of similar types of 2D images and for the acceleration of following computations.

1 Introduction

The acquisition and analysis of the visual information represents a powerful tool for interpretation of a large volume of input data. Recently, the origin of computer vision is intimately intertwined with computer history, having been motivated by a wide spectrum of important applications such as robotics, biology, medicine, industry and physics, and also in agricultural and food sciences. Among all different aspects underlying visual information, the shape of the objects certainly plays a special role. The multidisciplinarity of image analysis, with respect to both techniques and applications, has motivated a rich and impressive set of information resources represented e.g. in a book by Costa and Cesar[5].

This paper presents a completely different approach, where input image data are significantly reduced (to 5 % of original extent) by means of MAPLE 13 algorithm without any loss of precision. An example of this is a digital photo of carrot. Reduced data sets can be subsequently used for faster processing. The MAPLE software environment have been successfully used to determine the shape of agricultural products [1], [2], [3], [4].

2 Material and methods

2.1 Digital photo processing

A sample digital photo of carrot (bought in May 2010 in Kaufland, Jičín) has been used in this study. But any similar object of natural of artificial origin could

*The research has been supported by the Grant Agency of the Czech Academy of Sciences under Contract No. IAA201990701.

be used. The photo was taken by a digital camera Panasonic DMC-T27 with the resolution of 10.5 Mpixels. Points creating carrots perimeter were extracted and approximated as a polygon. This process is in detail described in earlier paper [4] and the applied Maple algorithm may be downloaded from author's web page: *www.user.mendelu.cz/barton*

2.2 Input data file organization

The input file contains three variables. The first one, **O** is a list of coordinates of N points describing carrot perimeter, $O_i = [O_{i_1}, O_{i_2}]$, $1 \leq i \leq N$. The second one **P** is a list of n vertexes of the polygon approximating carrot perimeter, $P_i = [P_{i_1}, P_{i_2}]$, $1 \leq i \leq 1$. List $\mathbf{\Lambda}$ is a list of n sublists containing coordinates of perimeter points corresponding to sides of the approximating polygon, $\Lambda_i = [P_{j_1}, P_{j_2}]$, $1 \leq j \leq n_i$. For example, Λ_k is k^{th} element of the $\mathbf{\Lambda}$ and contains coordinates of the perimeter points corresponding to k^{th} side of the polygon. This side is represented by a k^{th} line segment with endpoints P_k and P_{k+1}. All coordinates are in pixels.

2.3 Optimization

Each lateral side of the approximating polygon, hereinafter mentioned only as side, is given by the pair of their end points –$P1$ and $P2$. Square of distance of the point O from the line given by points $P1$ and $P2$ can be expressed as a function:

$$S(O, P1, P2) = \frac{(P2_1P1_2 - P2_1O_2 - P2_2P1_1 - P1_2O_1 + P1_1O_2 + P2_2O_1)^2}{(P1_1 - P2_1)^2 + (P1_2 - P2_2)^2} \ . \quad (1)$$

Sum of squares of distances of points corresponding to k^{th} side is:

$$q_k = \sum_{i=1}^{n_k} S(\Lambda_{i_k}, P_k, P_{k+1}) \ . \quad (2)$$

Finally sum of squares of all distances is:

$$Q = \sum_{k=1}^{n} q_k = \sum_{k=1}^{n} \left(\sum_{i=1}^{n_k} S(\Lambda_{i_k}, P_k, P_{k+1}) \right) \ , \quad (3)$$

where Λ_{i_k} is i^{th} member of the k^{th} sublist of the list $\mathbf{\Lambda}$; in other words, it is i^{th} point of the sublist Λ_k, corresponding to the k^{th} side. As we can see, $Q = Q(P1, \cdots, Pn) = Q(\mathbf{P})$ is a function only of coordinates of the approximating polygon, because coordinates of perimeter points are constant.

2.3.1 Global optimization – global data shaking

The approximating polygon is a closed curve; each endpoint of it belongs to two sides, $P1 = P_i$, $P2 = P_{i+1}$, for that reason it is not possible to optimize each side separately. We have to minimize Q with respect to $P_{n+1} = P1$, reflecting condition of the closed curve - approximating polygon, the end point of the last side is the first point of the first side. Because Q is a non-linear function of **P**, it was necessary to

use the iteration method. The Gauss-Newton iteration method is one of the most effective tools.

We have to find a new polygon with h vertexes saved in he vector - list $\mathbf{G} = [G_i]$, $1 \leq i \leq h$ minimizing $Q = Q(\mathbf{G})$. Vertexes saved in the list $\mathbf{P}$ may be used as an initial approximation. It is not necessary to put $h = n$, because during the iteration, both endpoints of the side may be so close that it will be possible to substitute them by one point. So we may assume that at the beginning of the iteration $h = n$,, and that later on it may be that $h \leq n$.

Vector $\mathbf{P}$ can be corrected by means of list of small corrections $\mathbf{\Delta P}$. In this case we can use:

$$Q(\mathbf{P} + \mathbf{\Delta P}) = Q(\mathbf{P}) + J_Q \, \mathbf{\Delta P} \ , \tag{4}$$

where J_Q is the Jacobian matrix and minimizing of (3) is converted into a linear problem of computation of the vector $\mathbf{\Delta P}$. Now we can put $\mathbf{P} = \mathbf{P} + \mathbf{\Delta P}$ and repeat the whole process until the moment when the requested accuracy is reached. The usual condition of accuracy is $||\mathbf{\Delta P}||_2 \leq \epsilon$, where ϵ is accuracy.

However this approach is divergent, and for that reason unusable.

2.3.2 Local optimization – local data shaking

The main idea of this approach is to optimize only the side with the largest distance between perimeter points and corresponding polygon sides. In this case we shall move only with two consequent $P1$ and $P2$ points from the vector $\mathbf{P}$, $P1 = P_i$, $P2 = P_{i+1}$. Index i corresponds to the side with the largest distance from the perimeter points.

We have to remember that we shall move with three sides. These sides have indexes $i - 1$, i and $i + 1$, and they are given by endpoints P_{i-1}, P_i, P_{i+1} and P_{i+2}, but points P_{i-1} and P_{i+2} are stable, without computed corrections. This approach is based on the same theory as global optimization, but with a reduced volume. Because $P1 = [P1_x, P1_y]$ and $P2 = [P2_x, P2_y]$ vector $\mathbf{P}$ may be organised as $\mathbf{P} = [P1_x, P1_y, P2_x, P2_y]$; organisation of vector of corrections is equivalent.

Non-zero elements of the Jacobian matrix J_Q corresponding to the first iteration step, $i = 4$, are displayed in the Fig. 1.

During the iteration the following cases may occur:

1. The simplest one is a convergence to desired accuracy. In this case there are no changes between points corresponding to sides $i - 1, \cdots, i + 1$.
2. If correction $\mathbf{P} = \mathbf{P} + \mathbf{\Delta P}$ is introduced, points on the perimeter may be closer to the other side. Points form $i - 1^{th}$ side may move up to i^{th} side, from $i + 1^{th}$ side may move down to i^{th} side. Points from i^{th} side may move down as well as up. This leads to a redistribution of points between sublists Λ_{i-1}, Λ_i and Λ_{i+1}.
3. If perimeter points are redistributed, the number of points corresponding to one side may be smaller than or equal to 3. In this case these points may

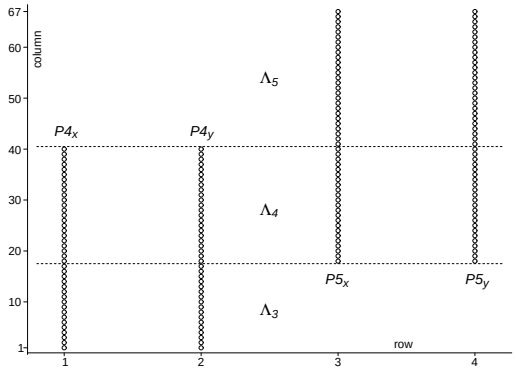

Fig. 1: *Non-zero elements of the J_Q corresponding to the first iteration step.*

be distributed between adjacent lines. The number of approximating polygon vertexes drops down by 1, $n = n - 1$. The iteration must be restarted.

4. Lines $i - 1$, i or i, $i + 1$ may be parallel. Line i is assumed to be parallel if $S(P_i, P_{i-1}, P_{i+1}) \leq 0.25$, similar condition may be used for lines i and $i + 1$. In this case these lines may be substituted by one with the endpoints P_{i-1}, P_i, or P_i, P_{i+1}, and sublists Λ_{i-1}, Λ_i or Λ_i, Λ_{i+1} may be collected. The number of approximating polygon vertexes drops down by 1, $n = n - 1$. The iteration must be restarted.
5. The point is jumping. In the iteration process the point jumps up and down. In this case the greatest difference of the optimized side of polygon is smaller than the second one of all points. The iteration may be finished.

If the optimization is finished, the whole process may be repeated with a new greatest distance until the moment when the same point will be after iteration again the point with the largest distance. In this case we have two variants of continuation:

Variant 1: To continue with the point with the second largest distance, later with third etc.

Variant 2: To put new polygon vertex into the point with the greatest distance and to split corresponding subvector L_i into two and to restart the whole process of iteration. The number of approximating polygon vertexes rises up by 1, $n = n + 1$.

The usual maximal distance is close to 1 pixel. For that reason it is not necessary to use Variant 2 very often, because the precision of digital photo is $\pm$ 1 pixel. This means that Variant 2 is used only from time to time.

3 Results

The result is again a polygon with a lower number of vertexes than in initial polygon and with a better approximation of the perimeter of the object. Quality of the approximation may be evaluated in the following ways:

1. By means of the greatest displacement.
2. By means of the average displacement.
3. By means of the number of polygon vertexes.
4. By means of the coefficient of linear correlation. The linear correlation is computed for vectors of distances from the origin of perimeter points and corresponding points on the polygon vertexes.

Results are presented in Tab. 1 and Figs. 2 and 3 demonstrating the optimization of the carrot digital photograph. Carrot perimeter creates 1819 vertexes.

Parameter	**Input polygon**	**Optimized polygon**
Greatest displacement	1.66	1.39
Average displacement	0.46 ± 0.33	0.38 ± 0.27
Vertexes	61	48
Correlation	0.9999923	0.9999942
Data reduction	3.35%	2.64%

Tab. 1: *Results of the optimization.*

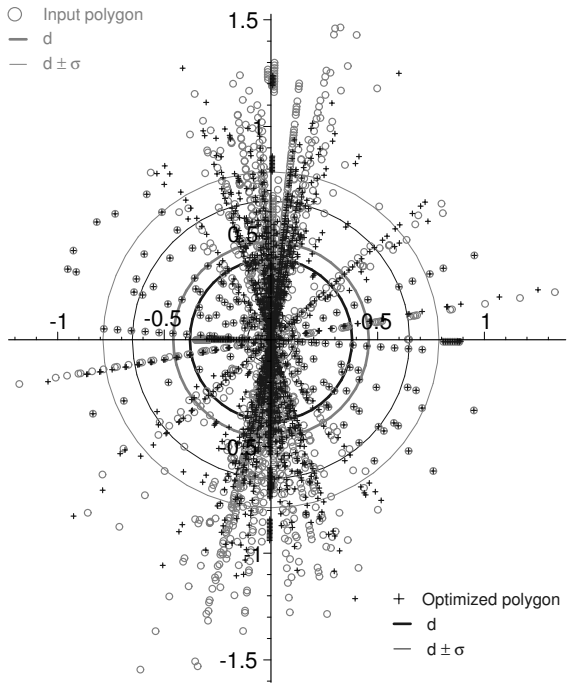

Fig. 2: *Vectors of displacements of the initial and optimized polygon. d = mean displacement, σ = quadratic error of the displacement.*

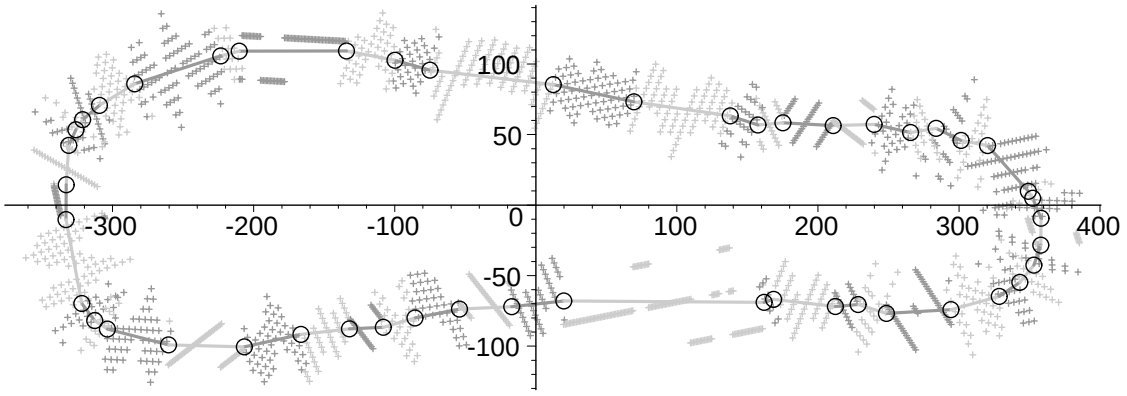

Fig. 3: *Visualisation of the optimized polygon. Displacements are 30× enlarged.*

4 Conclusions

The proposed procedure is of a general nature and can be used for data reduction for the evaluation of other biological as well as artificial shapes. It can serve as an effective and precise tool for acceleration of the process of computing and for enabling the calculation itself, when using less powerful hardware, e.g. common PC with a computer algebra program and/or in case of data processing using methods of non-linear regression.

References

[1] Bartoň, S.: Quick algorithms for calculation of coefficients of non-linear and partially continuous functions using the Least Square Method, solution in Maple 6. In: *Proceedings of 8th International Research Conference CO-MAT-TECH 2000*, pp. 79-85. MTF Trnava, STU Bratislava, 2000, ISBN 80-227-1413-5.

[2] Bartoň, S.: Stanovení tvaru zemědělské plodiny. In: *Proceedings of 6. Matematický workshop. Brno*, pp. 1-12. FAST VUT Brno, 2007, ISBN 80-214-2741-8.

[3] Bartoň, S.: Three dimensional modelling of the peach in Maple. In: Chleboun J., (Ed)., *Programs and Algorithms of Numerical Mathematics*, pp. 7-14. 1st ed. Praha. Matematický ústav AV ČR, 2008, ISBN 978-80-85823-55-4.

[4] Bartoň, S., Severa, L., and Buchar, J.: New algorithm for biological objects' shape evaluation and data reduction. In: *Acta of Mendel University of Agriculture and Forestry Brno*, vol. 58, No. 1, pp. 13–20. MZLU Brno, 2010, ISSN 1211-8516.

[5] Costa, L.F. and Cesar, R.M.: *Shape classification and analysis theory and practice*. CRC Press, 2009, ISBN 978-0-8493-7929-1.