Algorithm of dynamic programming for optimization of the global matching between two contours defined by ordered points

Francisco. P. M. Oliveira[1] and João Manuel R. S. Tavares[1]

Abstract: This paper presents a new assignment algorithm with order restriction. Our optimization algorithm was developed using dynamic programming. It was implemented and tested to determine the best global matching that preserves the order of the points that define two contours to be matched. In the experimental tests done, we used the affinity matrix obtained via the method proposed by Shapiro, based on geometric modeling and modal matching.

The proposed algorithm revealed an optimum performance, when compared with classic assignment algorithms: Hungarian Method, Simplex for Flow Problems and LAPm. Indeed, the quality of the matching improved when compared with these three algorithms, due to the disappearance of crossed matching, which is allowed by the conventional assignment algorithms. Moreover, the computational cost of this algorithm is much lower than the ones of other three, leading to enhanced execution times.

Keywords: Image analysis, contours matching, optimization, dynamic programming.

1 Introduction

The recognition of objects represented in images is one of the central problems in Computational Vision. It is a challenging task, mainly due to the large number of variations of projection of objects in 2D images, for instance as a result of changes of camera position, or even due to the deformations that the objects might have suffered.

To measure the similarity or disparity between two objects represented in images, or the same object represented in different instants, techniques based on the signal that represents them could be used. In these techniques, the images are considered as a 2D signal that represents the gray level or color, for instance. In many of these methods, the well developed techniques of signal processing based on Fourier or wavelet transforms are used. In [Daugman (2003)] a method is presented based in Gabor wavelets widely used for identifying people by their iris. Fourier or wavelet transforms are also frequently used in applications of image compression. In [Zhang (2007)] a comparative study is made among some methods based on wavelet transform for image compression.

A problem tightly related with the one of recognition of objects in images is the problem of identifying corresponding elements between images, often defined by groups of points, segments of straight lines or curves and boundaries. Frequently, these groups of points or segments represent the external contour of shapes represented in the input images.

To extract points, segments or boundaries of shapes represented in images it is necessary to divide the input image into regions. This process is usually called segmentation. Many segmentation techniques exist, such as, methods based on templates matching; statistical modeling; deformable templates; deformable models; level set methods, [Wang, Lim, Khoo and M. Wang (2007)] and physical principles, [Gonçalves, Tavares and Natal (2008)]. For a review of these methods see, for example, [Zhang (2001)] and [Tavares *et al* (2007)].

Different segmentation methods are applied to distinct situations to solve the image processing issue. As examples of application of these techniques, see for instance, [Zhang *et al* (2008)] and [Tavares *et al* (2007)].

The problem of finding correspondences among characteristic points of an object in two different instants, or between two objects, represented in images, originated the emergence of many proposals, in the sense of reaching the best global correspondence among the referred points. To determine the matches, the following techniques can be used: spatial intensity gradient information, [Lucas and Kanade

[1] FEUP – Faculdade de Engenharia da Universidade do Porto
INEGI – Instituto de Engenharia Mecânica e Gestão Industrial, Portugal
Email: franciscopmoliveira@yahoo.com.br, tavares@fe.up.pt

(1981)]; modal matching, [Shapiro and Brady (1992); Sclaroff and Pentland (1995)]; shape context, [Belongie, Malik and Puzicha (2002)]; curvature information, [Oliveira and Tavares (2007); Oliveira (2008)]; or distance to the center of the objects, [Oliveira (2008)].

When the similarity among characteristic points is quantified in a cost matrix, traducing the match affinities, the matching problem can be interpreted as an optimization problem. Usually, assignment algorithms are used to determine the best global matching. Such algorithms are frequently based in: simulated annealing, [Starink and Backer (1995)]; linear or integer programming, [Bastos (2003), Bastos and Tavares (2004, 2006)]; bipartite graph matching, [Fielding and Kam (2000)]; convex optimization, [Maciel (2001)]; dynamic programming, [Scott and Nowak (2006)]; etc.

The optimization of correspondences between two closed contours defined by a set of ordered points is constrained by an important rule that should not be discarded: *the relative order of the points to be matched should be preserved to guarantee the coherence of the matching obtained, avoiding, like this, crossed matching*.

This problem of finding the global matching of minimum cost preserving the order of the points of the input shapes contours is not trivial, because there are different ordinations that define the same contour. The solution described in this paper is based on dynamic programming and is able to solve this problem in a simple and fast way.

To experiment and compare the developed new dynamic programming algorithm with other assignment algorithms, it was integrated in a computational platform, already developed, [Tavares (2000); Tavares, Barbosa and Padilha (2000, 2002)]. The results of the comparison with the Hungarian Method, Simplex for Flow Problems and LAPm are presented and discussed further in this paper. The cost matrices used for the comparison were obtained using the modal matching methodology proposed by Shapiro, which was already integrated in the referred platform, [Shapiro and Brady (1992); Tavares (2000); Tavares, Barbosa and Padilha (2000, 2002); Bastos (2003); Bastos and Tavares (2004, 2006); Tavares and Bastos (2005)]. However, another cost matrices could be used.

In this paper, after enumerating some previous works developed to find the best global matching between objects, the problem of searching for the best correspondence between two sets of ordered points that preserves the order defined is considered. Afterwards, comparative results between the developed algorithm and the classic assignment algorithms already referred are presented. The last section is dedicated to final conclusions and future work perspectives.

2 Previous work

This work comes in the sequence of the work presented in [Tavares (2000); Tavares, Barbosa and Padilha (2000)], in which methodologies for matching characteristic points of two objects in images were implemented, using physical modeling and geometric modeling, complemented with modal matching, [Shapiro and Brady (1992), Sclaroff and Pentland (1995)]. Thus, those methodologies were used to determine the matching between characteristics points from two shapes represented in images, through the construction of an affinity matrix. Afterwards, this cost matrix was used to search for the desired correspondences. The solution presented to establish the matching had a pure local nature, in the sense that two points were only matched if, for each one of them, the other point was the nearest in cost terms. This way, frequently happened that some points were not corresponded and sometimes crossed matching occurred, see Fig. 1.

Figure 1: Matching found between two contours ("*heart5*" and "*heart6*") using a local approach. These contours are defined by 81 and 83 points, respectively.

In [Bastos (2003)], the work previously done in [Tavares (2000); Tavares, Barbosa and Padilha (2000)] was complemented through the implementation of three global optimizations methods, aiming the determination of the desired matching. Thus, the problem of searching for the best global matching between two contours was formulated as a classic assignment problem and three algorithms traditionally employed to solve these kind of problems were used, [Dell' Amico and Tooth (2000)]: the usual Hungarian Method, [Hillier and Lieberman (1995)]; the Simplex for Flow Problems, [Löbel (2000)], and the LAPm, [Volgenant (1996)]. When those assignment algorithms were applied on the affinity matrices established using physical or geometric modeling, the experimental results obtained improved considerably in comparison with the ones obtained using the previous methodology based on pure local aspects, [Bastos (2003); Bastos and Tavares (2004, 2006); Tavares and Bastos (2005)].

As already referred, when the assignment algorithms were applied to match contours defined by ordered point sets, it was verified that sometimes the matching found appeared without sense, that is, the order of the points was not considered and, therefore, crossed matches were present, see Fig. 2. Thus, the work here presented had as main aim to develop an assignment algorithm that must preserve the predefined order of the points to be matched.

Figure 2: Matching found between the contours of Fig. 1 using global optimization.

3 Definition of the problem

Let us begin by defining what means, in this work, *relative order* and *absolute order* of the points that define a contour. In Fig. 3, the sequence of points shown can be defined as: 1, 3, 4, 6, 7, 9. This sequence is monotonous increasing. Considering the same figure, it can also be defined the sequence: 4, 6, 7, 9, 1, 3. However, the former is not monotonous.

Considering Fig. 3 as a closed contour, it can be observed that the above two sequences define the same contour. Their difference is only the initial point considered. In this paper, we will say that the first sequence preserves the absolute order, because it is monotonous increasing, and that the second one only preserves the relative order.

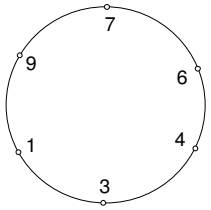


Figure 3: Sequence: 1, 3, 4, 6, 7, 9 placed on a circumference. The points of the circumference can also be defined, for instance, by the sequence: 4, 6, 7, 9, 1, 3.

To illustrate our solution for the problem of matching the points of two contours preserving their relative order, let us begin to analyze the two following examples:

a) Suppose that we have two contours, both defined by 4 points and numerated from 1 to 4, and consider the following matchings (given by column):

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} \text{ and } g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}.$$

When we observe the second line, which corresponds to the second contour, we can conclude that the matching f satisfies the absolute order but the matching g does not. However, the relative order is correct in both, because after point 1 comes point 2, after point 2 comes point 3 and so forth (considering the sequence of points disposed in circle).

b) Suppose now that we have two contours, one defined by 4 points and other defined by 7 points, respectively. Observe the next matchings:

$$h = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 5 & 7 \end{pmatrix},\ t = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 1 \end{pmatrix} \text{ and}$$

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 6 & 7 & 4 & 5 \end{pmatrix},$$

all of them preserve the relative order, but only matching h preserves the absolute order.

When the input contours are defined by equal number of points, the matching can be easily accomplished. In fact, it is enough to observe that if point i of contour 1 is matched with point j of contour 2, then point $i+1$ ($i+1$ means the point that follows point i in de sequence of points disposed in circle) of contour 1 has to be matched with point $j+1$ of contour 2, and so forth. Therefore, considering that both contours are defined by n points each one; there are just n hypotheses of global matching that preserve the relative order:

$$\begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 1 & 2 & 3 & \ldots & n \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 2 & 3 & 4 & \ldots & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 3 & 4 & 5 & \ldots & 2 \end{pmatrix}, \ldots, \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ n & 1 & 2 & \ldots & n-1 \end{pmatrix}.$$

Thus, it is enough to determine the cost of each one of the n global matchings and then choose the one that originated the minimum cost.

For contours defined by different number of points, we will present, afterwards, a new formulation based on dynamic programming, which finds the best global matching maintaining the absolute order of the matched points.

4 Formulation as a dynamic programming problem

4.1 General formulation

Let us begin this section considering a straightforward example. Let us suppose that we have contour 1 and contour

2 defined, respectively, by 4 and 6 points and the following cost matrix of the matches between them:

$$C = \begin{bmatrix} 1 & 0 & 1 & 4 & 5 & 1 \\ 0 & 3 & 1 & 5 & 2 & 1 \\ 6 & 1 & 2 & 4 & 0 & 8 \\ 3 & 2 & 7 & 5 & 4 & 1 \end{bmatrix},$$

where c_{ij} represents the cost to match point *i* from contour 1 with point *j* from contour 2.

To avoid crossed matches, we require that the absolute order of the matched points must be preserved. Thus, we impose the monotony of the matching sequence, that is, if point *i* of contour 1 is matched with point *j* of contour 2, then point $i+1$ of contour 1 has to be matched with a point $j+k$ of contour 2, where *k* is an integer and not less than one. Hence, we have, for instance, among others, the following valid matchings:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 5 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 6 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 5 & 6 \end{pmatrix},$$

with the associated global costs: 11, 10, 6 and 7, respectively.

In total, for the imposed hypotheses, we have exactly 15 possible global matchings, because to count the global matching hypotheses is equivalent to count how many subsets of 4 different elements we can get from the 6 elements of contour 2. Therefore, the number of global matchings that preserve the absolute order is, in this example, given by:

$$C_4^6 = \frac{6!}{(6-4)!2!} = 15.$$

In general, if a contour is defined by *n* points and the other by *m* points, with $n \le m$, there are exactly C_n^m (combinations of *n* elements in a set of *m* elements) matching hypotheses maintaining the absolute order. Considering the relative order, there are exactly mC_n^m hypotheses, as we will explain later.

Using a usual notation in dynamic programming, [Norman (1975) and Winston (1994)], for the previous example, we will define 4 stages. In stage 1, the match of smaller cost for point 1 of contour 1, under the matching hypotheses is chosen. In stage 2, the best match for point 2 of contour 1 is selected, under the matching hypotheses derived from the match of point 1 in stage 1, and so forth. It is fundamental to refer that the definition of a match between two points in a certain stage will affect the hypotheses of matching in the subsequent stages.

For better understanding of the former approach, let us observe the following. In the example in study, point 1 of contour 1 can just be matched with points 1, 2 or 3 of contour 2, but, for instance, if point 1 of contour 1 is matched with point 3 of contour 2, then point 2 of contour 1 has only one matching hypothesis: with point 4 of contour 2. Thus, and according to the matching already done in the previous stages, for a certain stage *k* from the example in study, point *k* of contour 1 will be matched with only a point of the following groups of points of contour 2:

$\{k\}$, $\{k, k+1\}$ or $\{k, k+1, k+2\}$.

To indicate how many points of contour 2 are available to be matched with a certain point of contour 1, we will define the state variable *s*. For the referred example, we have $s \in \{1,2,3\}$. If in a certain stage *k* we have $s = 1$, then point *k* of contour 1 has only one matching hypothesis (with point *k* of contour 2); if $s = 2$, then point *k* of contour 1 has two matching hypotheses (with points *k* or $k+1$ of contour 2), and so on.

Let us now define the function of minimum cost $f_k(s)$, where *s* is the state variable already defined, *k* represents the stage and $f_k(s)$ represents the minimum cost to match points 1, 2, 3… *k* of contour 1, when point *k* of contour 1 has *s* matching hypotheses of choice.

To better elucidate our approach, we will apply this formulation on the example in study. Thus, we will build, successively, an optimal matching that preserves the absolute order of the points involved. For such, on the left we indicate the minimum costs for each stage and for each state, and on the right we define the matching established:

$$f_1(1) = c_{11} = 1 \qquad \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$f_1(2) = \min\{c_{11}, c_{12}\} = 0 \qquad \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$f_1(3) = \min\{c_{11}, c_{12}, c_{13}\} = 0 \qquad \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$f_2(1) = c_{22} + f_1(1) = 3 + 1 = 4 \qquad \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

$$f_2(2) = \min\{c_{22} + f_1(1), c_{23} + f_1(2)\} = 1 \qquad \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

…

$$f_4(3) = \min\{...\} = 2 \qquad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 5 & 6 \end{pmatrix}$$

As in total there are 4 stages, if the intension is just to calculate the minimum cost, in the fourth stage it would not be necessary to calculate $f_4(1)$ and $f_4(2)$ but, because it is necessary to keep relative information for the matching, such calculations have to be done. For the example in study, the minimum cost to match the 4 points of contour 1 with 4 points of contour 2, preserving the absolute order of the points, is equal to 2 and the associated matching is the last one.

In general, for a cost matrix C of dimension $n \times m$, with $n \le m$, $k \le n$ and $s \in \{1, 2, ..., m-n+1\}$, $f_k(s)$ represents the minimum cost to match the points 1, 2, …, k of contour 1, when point k of contour 1 has s matching hypotheses. With this formulation, we guarantee that the best global matching that preserves the absolute order is reached.

To obtain the best global matching while maintaining the relative order, it is necessary to rearrange the points of contour 2 (point 2 becomes point 1; point 3 becomes point 2 and so forth). Continuously, the matching of minimum cost that preserves this new absolute order and the respective cost are calculated. The rearrangement process and consecutive calculus are repeated again, and so forth.

With the described approach, each new absolute order corresponds to a relative order, relatively to the initial arrangement. Thus, all of the possible relative arrangements of contour 2 are built, and thus all the matchings that preserve the relative order and respective minimum costs are obtained.

In the example in study, it is necessary to solve 6 problems of global matching that preserve the new successive absolute arrangements of the points of contour 2. After applying this formulation, the matching of minimum cost that preserves the relative order of the points still is the previously presented.

4.2 Algorithm and implementation

Before we present our new algorithm, let us observe the example described in the previous section. In that example, we have, for instance:

$$f_3(3) = \min\{c_{33} + f_2(1), c_{34} + f_2(2), c_{35} + f_2(3)\}.$$

It seems that to calculate $f_3(3)$ we have to calculate three values and later compare them to choose the lower one. However, such procedure is not necessary, because the values $c_{33} + f_2(1)$ and $c_{34} + f_2(2)$ were already calculated and $c_{34} + f_2(2) \le c_{33} + f_2(1)$. According to this, it is enough to calculate $c_{35} + f_2(3)$ and compare it with $c_{34} + f_2(2)$. Thus, in each stage, only one sum operation and one comparison operation for each state is done, if $s > 1$. If $s = 1$, then only one sum is necessary.

The presented algorithm starts from the hypothesis that it is not known a *priori* any matches that should be considered. For that reason, it determines all the possible global matchings that preserve the new successive absolute orders and then it chooses the one of minimum cost. The chosen matching is the one of lower cost that maintains the relative order of the points.

Our new algorithm can be described as follows:

Algorithm:

1. Read the dimension of contours to be matched and the costs matrix C. Define the value of n and m so that $n \le m$. If necessary ($n > m$), make the transpose of matrix C.
2. Repeat m times:
   i. To $k = 1,2,...,n$ and $s = 1,2,...,m-n+1$, calculate the values of $f_k(s)$, taking in consideration what was referred before, avoiding repeated calculations already made. Keep the values of $f_k(s)$ in a table of n rows and $m-n+1$ columns, that is, the used table must have so many rows as stages and so many columns as states, (Tab. 1).
   ii. Determine and keep the minimum cost, which is the value kept in the position $(n, m-n+1)$ of the values table. (In the previous example, it is the value kept in position $(4, 3)$ of Tab. 1).
   iii. Define and keep the global matching of minimum cost, which is made by making a search in the built table. Notice that the selection of a certain cell (i, j) means that the point i of contour 1 is matched with point $i + j - 1$ of contour 2. (See the cells used to define the matching in the example in study, Tab. 1.)
   iv. Rearrange the columns of the matrix C, so that, column 2 becomes column 1, column 3 becomes column 2 and so forth.
3. Seek the minimum cost between the m kept values and the respective matching.

If one match is known a *priori*, then the algorithm does not need to determine all the possible global matchings as in the presented case. For instance, let us suppose that it is known that point i of contour 1 should be matched to point j of contour 2. Then, the points of both contours are rearranged: point i of contour 1 becomes point 1, point $i+1$ becomes point 2 and so forth. The same is made in contour 2. Now, it is enough to solve only one problem to search for the best global matching that preserves the new absolute order, instead of m problems that the algorithm will have to solve if any match was known a *priori*.

Table 1: Minimum costs kept by the algorithm for the example in study. The values are relative to the first problem (initial order). The marked cells are used to define the matching.

	State (s)		
Stage (k)	1	2	3
1	$f_1(1)=1$	$f_1(2)=0$	$f_1(3)=0$
2	$f_2(1)=4$	$f_2(2)=1$	$f_2(3)=1$
3	$f_3(1)=6$	$f_3(2)=5$	$f_3(3)=1$
4	$f_4(1)=11$	$f_4(2)=9$	$f_4(3)=2$

4.3 Computational cost

Considering a contour defined by n points and another one defined by m points, with $n \le m$, for each global matching that preserves the absolute order there are n stages and $m-n+1$ states. For each stage, only one sum for state is effectuated. For each state larger than 1 (one) only one comparison is effectuated. Thus, we have in total $n\times(m-n+1)$ sums and $n\times(m-n)$ comparisons, considering only the fundamental operations involved.

To obtain the best global matching preserving the relative order, we have to solve m problems; therefore, there are $m\times n\times(m-n+1)$ sums and $m\times n\times(m-n)$ comparisons. To choose the best global matching from among all the global ones, we have more $m-1$ comparisons.

From the explained, we can conclude that execution time will increase when the number of points that define the contours increases and decreases when the difference among the number of points of the two contours decreases.

5 Dynamic programming with restriction of order *versus* Hungarian Method, Simplex for Flow Problems and LAPm

5.1 Test conditions

Before presenting some of the experimental results obtained, it is important to refer that this comparison was accomplished after the implementation of our new algorithm of dynamic programming in the computational platform for image processing and analysis already referred, [Tavares (2000), Tavares, Barbosa and Padilha (2002)]. To compare the two optimization methods – assignment algorithms without order restriction (AAWOR) and the dynamic programming algorithm with order restriction (DPAWOR) – one employed affinity matrices obtained using the methodology integrated in the same platform, based on geometric modeling and modal matching, proposed by Shapiro, [Shapiro and Brady (1992); Tavares (2000); Tavares, Barbosa and Padilha (2002); Bastos (2003); Bastos and Tavares (2004, 2006); Tavares and Bastos (2005)].

To compare the optimization algorithms based on the Hungarian Method, Simplex for Flow Problems and LAPm with the new optimization algorithm based on dynamic programming, it is necessary that the process to determine the cost matrix associated to the points that define both contours be exactly the same. Thus, in all of the experimental tests done, the configuration defined by default in the computational platform used for the building process of the affinity matrices was adopted.

In the definition of the Simplex for Flow Problems algorithm integrated in the computational platform adopted, the default configuration was also used, because it is, in general, the fastest, Fig. 4. To get the time required by each one of the optimization algorithms considered, a function already available for that proposed in the same platform was used.

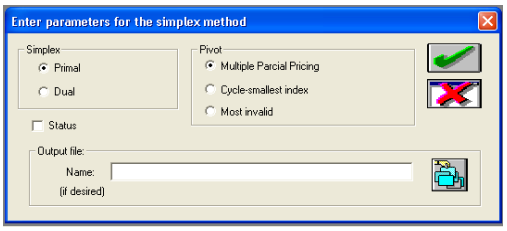


Figure 4: Configuration defined by default in the computational platform for the optimization algorithm based on the Simplex.

5.2 Results

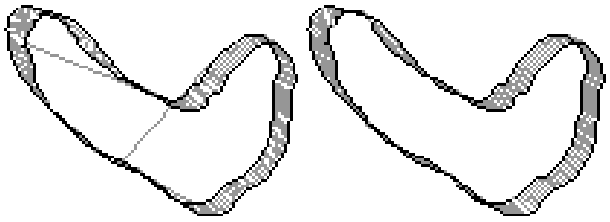
The quality of the matchings obtained using AAWOR and DPAWOR algorithms, in most of the contours tested, were exactly the same and excellent. The differences appeared when AAWOR presented crossed matches, what obviously did not happen with DPAWOR.

To illustrate the differences of the matches found by the two types of algorithms considered in some experimental cases, observe Figs. 2, 5, 6, 7, 8, 9 and 10. In those, the contours were aligned by applying the rigid transformation estimated. In some of the cases presented there are small differences in the positions of the contours, because the angle of rotation of a contour in relation to the other one is obtained based on the matches found. Thus, bad matches can originate an erroneous rotation angle.

In Tab. 2, we present the computational times required to determine the matching of several pairs of ordered contours and the respective matching costs. Some of the matching results indicated are not illustrated in this paper because they were equal for the two types of algorithms in comparison, or present almost imperceptive differences. It is important to

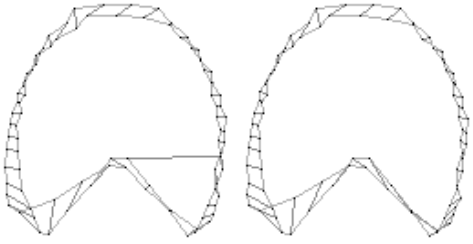
refer that the cost of the global matching relies on the elements of the cost matrix and that this one depends on the contours and the values of the parameters considered in the Shapiro's matching methodology. The time indicated is an average time, because small variations were observed. In several situations, the execution time was very low and for that reason the computational platform indicated an execution time of 0 (zero) seconds. Thus, in those situations we indicate in Tab. 2 a time "<0.01".

Figure 5: Matching found between the contours of Fig. 1 using the algorithm based on dynamic programming.



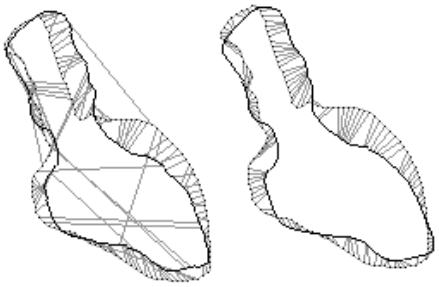
(a) (b)

Figure 6: Contours "*foot13*" and "*foot14*", defined by 233 and 253 points, respectively, and (a) matching found using AAWOR, (b) matching found using DPAWOR.



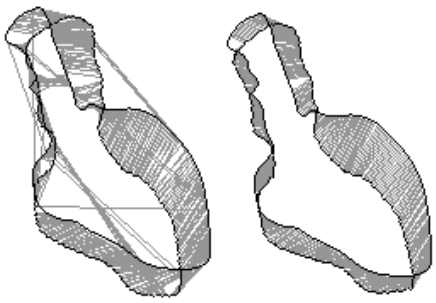
(a) (b)

Figure 7: Contours "*rib1*" and "*rib2*", both defined by 46 points, and (a) matching found using AAWOR, (b) matching found using DPAWOR.



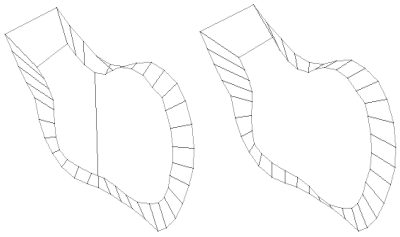
(a) (b)

Figure 8: Contours "*heartB3*" and "*heartB2*", defined by 389 and 139 points, respectively, and (a) matching found using AAWOR, (b) matching found using DPAWOR.



(a) (b)

Figure 9: Contours "*heartB3*" and "*heartB4*", defined by 389 and 417 points, respectively, and (a) matching found using AAWOR, (b) matching found using DPAWOR.



(a) (b)

Figure 10: Contours "*heartA1*" and "*heartA2*", both defined by 36 points, and (a) matching found using AAWOR; (b) matching found using DPAWOR.

6 Conclusions and future work perspectives

Relatively to the matchings found, the AAWOR algorithms always present, obviously, a solution of minimum cost, because they are driven by the same restriction. Besides, only in very singular situations more than one matching of minimum cost exists. Thus, the matchings obtained by the three assignment algorithms were always equal.

Table 2: Comparison between AAWOR and DPAWOR algorithms. (The experimental tests were accomplished using a PC Pentium III, at 1GHz with 256MB of RAM and Microsoft Windows XP.)

N. of points and "contour name"		Global cost of the matching		Execution time [s]			
Contour 1	Contour 2	Hung./Simp./LAPm	Dynamic	Hungarian	Simplex	LAPm	Dynamic
28, "*heart1*"	28, "*heart1a*"	0.00266	0.00266	4.286	0.02	0.01	<0.01
36, "*heartA1*"	36, "*heartA2*"	0.98965	1.1468	>60	0.04	2.352	<0.01
46, "*rib1*"	46, "*rib2*"	3.63974	4.06635	>60	0.06	2.774	<0.01
86, "*airplane12*"	57, "*airplane2*"	1.74522	1.74522	>60	0.20	1.332	0.01
81, "*heart5*"	84, "*heart6*"	5.79033	6.70609	>60	0.20	2.426	<0.01
233, "*foot13*"	67, "*foot2*"	6.0508	6.11264	>60	1.332	15.983	0.25
233, "*foot13*"	253, "*foot14*"	50.5486	57.9803	>60	2.013	>60	0.15
389, "*heartB3*"	139, "*heartB2*"	24.8986	25.7363	>60	5.418	>60	3.796
389, "*heartB3*"	417, "*heartB4*"	12.3774	13.833	>60	9.864	>60	1.192

The comparison between the results obtained using AAWOR algorithms and DPAWOR algorithm allows us to conclude the following:

- Whenever the AAWOR reached a good matching without crossed matches, the DPAWOR reached the same matching; therefore the global cost of the matching was exactly the same for the two types of algorithms.
- When the AAWOR reached a matching with some crossed matches, the DPAWOR reached an identical matching but without crossed matches. Obviously, the cost associated was superior because the restriction of the order forced some crossed matches to be substituted by matches of larger costs but more coherent.
- In the situations where the matching obtained by AAWOR were in the major part without sense, so were the matching obtained using DPAWOR. It is important to refer that those bad matchings were not due to the optimization algorithms used but to the methodology adopted in the construction of the cost matrix. Thus, no example of this situation was presented in this paper.

The execution time of the DPAWOR algorithm was always inferior to the execution time of all the AAWOR algorithms, independently of the contours have been defined by equal or different number of points, or if that number is high or low. Although the tests were executed in a slow computer, when compared with the more modern ones, there were situations in which the computational platform indicated execution times of 0 (zero) seconds for DPAWOR, what means a very low computational time.

It can be verified that the execution times of the DPAWOR algorithm varied in agreement with what was anticipated in section 4.3. In other words, the time increased when the number of points that define the contours increased, and it decreased when the difference between the number of points that define the two contours decreased.

Finally, as perspectives of future work, we hope to apply our DPAWOR algorithm to establish the matching of characteristic points of objects represented in images using several methodologies for the definition of the matching cost matrix, where the order of the points or other characteristics of the shape or image should be considered and preserved.

Acknowledgements

This work was partially done in the scope of project "Segmentation, Tracking and Motion Analysis of Deformable (2D/3D) Objects using Physical Principles", financially supported by FCT - Fundação para a Ciência e a Tecnologia in Portugal, with reference POSC/EEASRI/55386/2004.

References

Bastos, L. (2003): MSc Thesis: Optimização da determinação das correspondências entre objectos deformáveis no espaço modal. *Faculdade de Engenharia da Universidade do Porto*, Portugal.

Bastos, L.; Tavares, J. (2004): Objects matching improvement using optimization techniques in a geometric modal methodology". In: *WSCG'2004 - 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2004*, Plzen, Czech Republic.

Bastos, L.; Tavares, J. (2006): Matching of objects nodal points improvement using optimization. *Inverse Problems in Science and Engineering*. vol. 14, n. 5, pp. 529-541.

Belongie, S.; Malik, J.; Puzicha J. (2002): Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n. 24, pp. 509-522.

Daugman, J. (2003): The importance of being random: Statistical principles of iris recognition. *Pattern Recognition*, vol. 36, pp. 279-291.

Dell' Amico, M.; Tooth, P. (2000): Algorithms and codes ford dense assignment problems: the state of the art. *Discrete Applied Mathematics*, vol. 100, pp. 274-278.

Fielding, G.; Kam, M. (2000): Weighted matching for dense stereo correspondence. *Pattern Recognition 33,* n. 9, pp. 1511-1524.

Gonçalves, P.; Tavares, J.; Natal, R. (2008): Segmentation and simulation of objects represented in images using physical principles. *CMES: Computer Modeling in Engineering & Sciences.* (Accepted for publication)

Hillier, F.; Lieberman, G. (1995): Introduction to operations research. *McGraw-Hill International Editions.*

Löbel, A. (2000): MCF – A Network simplex implementation. http://www.zib.de/optimization/software/mcf (accessed in 2000)

Lucas, B.; Kanade, T. (1981): An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, April, 1981, pp. 674-679.

Maciel, J. (2001): PhD thesis: Global matching: optimal solution to correspondence problems. *Instituto Superior Técnico,* Lisboa, Portugal.

Norman, J. (1975): Elementary dynamic programming.

Eduard Arnold (Publishers), London.

Oliveira, F.; Tavares, J. (2007): Matching contours in images using curvature information. In: *VIPimage – I ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, pp. 375-379. Taylor & Francis.

Oliveira, F. (2008): MSc Thesis: Emparelhamento de objectos representados em imagens usando técnicas de optimização. *Faculdade de Engenharia da Universidade do Porto,* Portugal.

Sclaroff, S.; Pentland, A. (1995): Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, n. 6, pp. 545-561.

Scott, C.; Nowak, R. (2006): Robust contour matching via the order-preserving assignment problem. *IEEE Transactions on Image Processing*, vol. 15, n. 7, pp. 1831-1838.

Shapiro, L.; Brady, M. (1992): Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, vol. 10, n. 5, pp. 283-88.

Starink, J.; Backer, E. (1995): Finding point correspondences using simulated annealing. *Pattern Recognition*, vol. 8(2), pp. 231-240.

Tavares, J. (2000): PhD Thesis: Análise de movimento de corpos deformáveis usando visão computacional. *Faculdade de Engenharia da Universidade do Porto,* Portugal.

Tavares, J.; Barbosa, J.; Padilha, A. (2000): Matching image objects in dynamic pedobarography. In: *RecPad 2000 - 11th Portuguese Conference on Pattern Recognition*, Porto, Portugal.

Tavares, J.; Barbosa, J., Padilha, A. (2002): Apresentação de um banco de desenvolvimento e ensaio para objectos deformáveis. *RESI – Revista Electrónica de Sistemas de Informação*, Edição 1, vol. 1, n. 1.

Tavares, J.; Bastos, L. (2005): Improvement of modal matching image objects in dynamic pedobarography using optimization techniques. *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, pp. 1-20.

Tavares, J.; Carvalho, F.; Oliveira, F.; Vasconcelos, M.; Gonçalves, P.; Pinho, R. (2007): Computer analysis of Objects' movement in image sequences: methods and applications. In: *DSM 2007 – Conferência Nacional de Dinâmica de Sistemas Multicorpo*, ISBN: 978-989-20-0968-1, pp. 33-40, Universidade do Minho, Guimarães, Portugal, 6-7 Dezembro 2007

Tavares, J.; Bastos, L. (2005): Improvement of modal matching image objects in dynamic pedobarography using optimization techniques. *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, pp. 1-20.

Volgenant, A. (1996): Linear and semi-assignment problems: a core oriented approach. *Computers and Operations Research*, vol. 23, n. 10, pp. 917-932.

Wang, S.; Lim, K.; Khoo, B.; Wang, M. (2007): A Geometric Deformation Constrained Level Set Method for Structural Shape and Topology Optimization. *CMES: Computer Modeling in Engineering & Sciences*, vol. 18, n. 3, pp. 155-181.

Winston, W. (1994): Operations research: applications and algorithms, 3rd edition. *Duxbury Press,* USA.

Zhang, Y. (2001): A review of recent evaluation methods for image segmentation. In *Sixth International Symposium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia: IEEE, 2001.

Zhang, J. (2007): A Comparative study of non-separable wavelet and tensor-product wavelet in image compression. *CMES: Computer Modeling in Engineering & Sciences*, vol. 22, n. 2, pp. 91-96.

Zhang, Y.; Cheng, B.; Oh, C.; Spehar, J.; Burgess, J. (2008): Kinematic analysis of lumbar spine undergoing extension and dynamic neural foramina cross section measurement. *CMES: Computer Modeling in Engineering & Sciences,* vol. 29, n. 2, pp. 55-62.