Two-Dimensional Discrete Morphing*

Isameddine Boukhriss, Serge Miguet, and Laure Tougne

Université Lyon 2, Laboratoire LIRIS,
Bâtiment C, 5 av. Pierre Mendès-France,
69 676 Bron Cedex, France
{iboukhri, smiguet, ltougne}@liris.univ-lyon2.fr
http://liris.cnrs.fr

Abstract. In this article we present an algorithm for discrete object deformation. This algorithm is a first step for computing an average shape between two discrete objects and may be used for building a statistical atlas of shapes. The method we develop is based on discrete operators and works only on digital data. We do not compute continuous approximations of objects so that we have neither approximations nor interpolation errors. The first step of our method performs a rigid transformation that aligns the shapes as best as possible and decreases geometrical differences between them. The next step consists in searching the progressive transformations of one object toward the other one, that iteratively adds or suppresses pixels. These operations are based on geodesic distance transformation and lead to an optimal (linear) algorithm.

1 Introduction

Many medical images are produced every day and their interpretation is a very challenging task. 3D atlases can be of great interest since they allow to help this interpretation by very precise models. Most of the time, these atlases are built manually and represent a considerable amount of work for specialists of the domain. Moreover, they only contain static information corresponding to a single patient or potientially an average shape corresponding to a small set of patients. It would be very useful to compute these atlases in an automated way from a set of images: it would allow to compute not only an average shape between all input data but also statistical measures indicating the interindividual variability of these shapes. This is the basic idea of the statistical atlas [FLD02].

In this paper, our goal is to study the progressive deformation from one object to another one which is the first step for the computation of an average object. For sake of simplicity, we focus on 2D binary images but the proposed approach could easily be generalized to 3D. Our method is decomposed into two steps: the first one consists in making a rigid registration of the two objects and the second one in computing the deformation.

* This work was supported by the RAGTIME project of the Rhône-Alpes region.

R. Klette and J. Žunić (Eds.): IWCIA 2004, LNCS 3322, pp. 409–420, 2004.

2 State of the Art

Concerning the rigid registration, many algorithms exist in the literature. Some of them are based on intensity and they use similarity measures such as correlation coefficients, correlation ratios [RMP+98] and mutual information [WV96]. These algorithms do not need segmentation of the images and are based on the statistic dependences between the images to be registered. Other algorithms are based on geometrical properties such as surfaces, curvatures [MMF98] and skeletons [LB98]. These methods are generally faster than the previous ones, nevertheless less precise. As a matter of fact, the extraction of the surface and the computation of the curvature are noise sensitive and they may induce imprecision in the registration. A discrete method has been proposed by Borgefors [Bor88] which uses distance transforms. In this method, for one of the images the associated distance card is computed. The object of interest in the second image is approximated by a polygon which is superimposed on the previous distance card. Then, the squared values of the pixels of the distance card in which the polygon is superimposed are averaged to obtain a contour distance. The author searches for the rigid transformation of the polygon that minimizes this distance. However, it is difficult to estimate the number of necessary iterations.

We can also find lots of methods in the literature that make the morphing of two shapes. An important part of them is based on the interpolation of the positions and/or the colours of the pixels in the two images [Iwa02]. In our case, we consider the morphing as generation of intermediate images.

A recent study [BTSG00] computes the average shape between two continuous shapes. First, it makes the registration of the two images and it computes the skeleton of the difference between the two shapes. Using an elimination process, it only keeps the points of the skeleton that are equidistant of two borders of two different objects. This method preserves the topology and the initial shape of the objects. However, it only allows to generate the median shape and not a continuous deformation of one shape to the other one. The method we present in this paper is a generalized discrete version of the previous one. The generalization we propose allows to compute not only a median shape but also the different intermediate shapes.

In the following section we recall some notions necessary for the comprehension of the remainder of the text. Section 4 is the heart of the article and describes the proposed method. In section 5, we present some examples in which we have applied our method. Finally, we conclude and present some future works.

3 Preliminaries

Let us give the formal context of our study and recall some basic notions concerning the inertia moments.

3.1 Neighborhood, Connectivity and Distance

We consider 2D shapes in the $\mathbb{Z}^2$ space. The pixels of the shape have the value 1 and the pixels that belong to the background have the value 0. The object is considered as 8-connected and background is considered as 4-connected. We work in 3×3 neighborhood. Let a and b denote two binary shapes, we denote the symmetric difference by $a \Delta b = \{a \cup b\} \backslash \{a \cap b\}$.

In the following we will use the distance transform. This is a classical tool [RP68] which associates to each point of the object the distance of the nearest pixel of the background. In our study, we will use the chamfer distance 3-4 which is a good approximation of the Euclidean distance. The figure 3.1 gives an example of a distance transform obtained in this way.

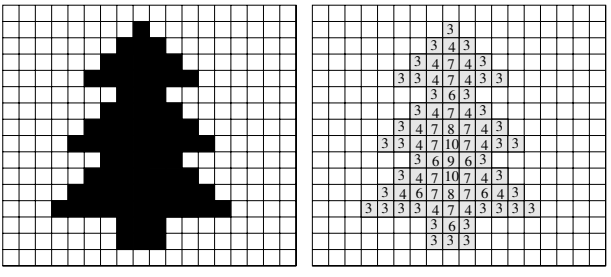

Fig. 1. An example of distance transform using the chamfer 3-4 distance

3.2 Inertia Moments

In order to make the registration, we will use the moments associated to the shapes. Such descriptors are especially interesting in order to determine the position, the orientation and scale of an object. Moreover, one of their main advantages is their small sensitivity to noise.

Let us consider a two-dimensional image I. The general form of the discrete moments is:

$$M_{pq} = \sum_x \sum_y (x - X_c)^p (y - Y_c)^q I(x, y)$$

with $0 \leq p, q \leq 2$.

X_c and Y_c are the coordinates of the barycentre of the shape :

$$X_c = \frac{1}{N} \sum_i X_i \qquad\qquad Y_c = \frac{1}{N} \sum_i Y_i$$

From these moments we can compute the principal inertia axis of the shape. Such a computation is made with the help of the inertia matrix:

$$MI = \begin{pmatrix} M_{20} & -M_{11} \\ -M_{11} & M_{02} \end{pmatrix}$$

This matrix is normalized and diagonalized in order to obtain the eigenvectors V_1 and V_2 and the associated eigenvalues λ_1 and λ_2. Let us suppose $\lambda_1 > \lambda_2$. V_1 represents the maximal elongation axis of the object.

In the following section, we describe the proposed method to obtain all the intermediary images between two given images.

4 Methodology

The method is based on an operation that consists in aligning the two figures. This operation is described in subsection 4.1. When the two shapes are superimposed, in the same referential, we can start the morphing step. The morphing operation is described in subsection 4.2.

4.1 Rigid Registration

The rigid transformation consists in a sequence of global operations applied on the shapes in order to superimpose the shapes in the same lattice. We can decompose such a sequence into two parts: the first one is the rigid registration itself and the second one deals with scaling and re-sampling.

In order to show the different steps, we have chosen two shapes, presented in the figure 4.1, representing fishes we want to align.

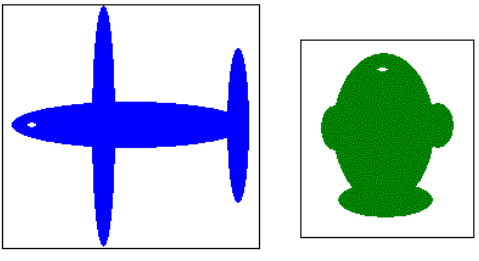

Fig. 2. Two shapes we want to align

The first step is directly deduced from the computation of the inertia moments presented in subsection 3.2 and consists in a translation and a rotation. Simply, it consists in computing the inertia moments of order one and two for each shape. It is then followed by the application of Backward Mapping [Wol90] to make the transformation. Figure 3 gives the resulting image.

The scaling and re-sampling operations are very important because we cannot compare two shapes that are not digitalized on the same grid. Our goal is to make homothetic scaling in order to preserve the global aspect of the shapes. An intuitive method would be to decrease the size of the biggest shape or to increase the size of the smallest one. In fact, it would be equivalent in a continuous domain. But, in the discrete space the objects can have different samplings and if we reduce the size we may lose important information; on the contrary, if

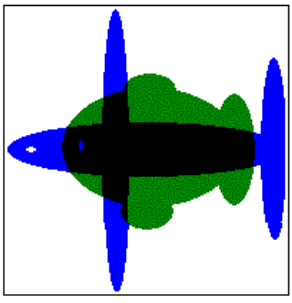

Fig. 3. Rigid registration: translation and rotation

we zoom a digital object we may generate aliasing artefacts. We have chosen a compromise which aligns the principal vectors on a new system of coordinates and which chooses an intermediate scale between the two shapes (it reduces the biggest and increases the smallest). Moreover, as we have mentioned previously, the two shapes may not have the same sampling resolution: the pixel size along the x axis may not be the same as along the y axis and, they also may be different between the two images. This is an other reason why we do not re-sample only one of the shapes using the grid of the other one but prefer to use a third lattice that is good adapted to the re-sampling of the two shapes. Figure 4 shows the resulting image after the scaling and re-sampling operations in our example.

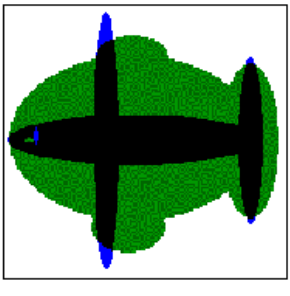

Fig. 4. The two shapes after registration, scaling and re-sampling

Note however that we have shown here the different steps in order the make it more easily understandable, but in fact all the corresponding transformation matrices are combined together to obtain only one transformation matrix T that is applied at once. For classical reasons linked to discrete transformations using different lattices, we have chosen to use the backward mapping technique to apply the transformation. Using bounding boxes of the input shapes, we determine the limits of the output space and for each pixel of the output space we compute its predecessor using T^{-1}. Even if this technique may generate aliasing

we will see in the final results that it will be negligible for the conservation of the global aspect of shapes.

The principal axis is a direction of the maximal elongation of a shape. The alignment of the principal axis we have described in this section might be subject to orientation errors. Among the two possible orientations we have for direct transformations, we select the one that maximizes the Hamming distance between the two shapes.

4.2 Discrete Morphing

The two shapes are now discretized in the same lattice. We want to define a transformation that progressively deforms one of them into the second. Just remark here that the transformation we propose is symmetrical: we can choose as first image indifferently one of the two images. In order to obtain the intermediate shapes, we iteratively add pixels to shape a (those belonging to $b \backslash a$) and delete other pixels from shape a (those belonging to $a \backslash b$) until the object converges to shape b. The decision concerning the adding or deleting is determined by the distance information. As a matter of fact, the first step consists in assigning to the symmetric difference of the two shapes a distance information. In a second step, we make a dilation or an erosion of the difference according to this distance information. However the propagation speed cannot be the same in all directions: the longer the contour of a has to progress toward the contour of b, the faster has to be the propagation speed. We have thus to compute the different 4-connected components of $a \Delta b$ (see figure 5) and to set up the propagation speed in each component proportional to the largest distance information in this component (see details below).

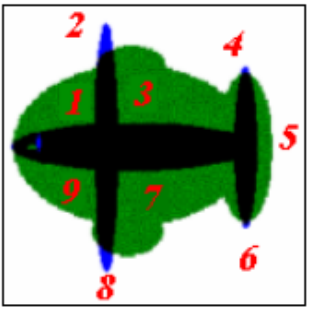

Fig. 5. Connected components labelling

Geodesic Propagation. Using the distance cards described in subsection 3.1, we construct two kinds of geodesic waves: the first one is initialized by the intersection of the two shapes and propagates in the difference of the two shapes toward the background. The second one is initialized by the background and propagates in the difference toward the intersection. We denote by d_1 the 3-4 distance from

$a \cap b$ to $a\Delta b$, and d_2 the 3-4 distance from the complement of $a \cup b$ to $a\Delta b$. The figure 6 shows the two geodesic waves in our example.

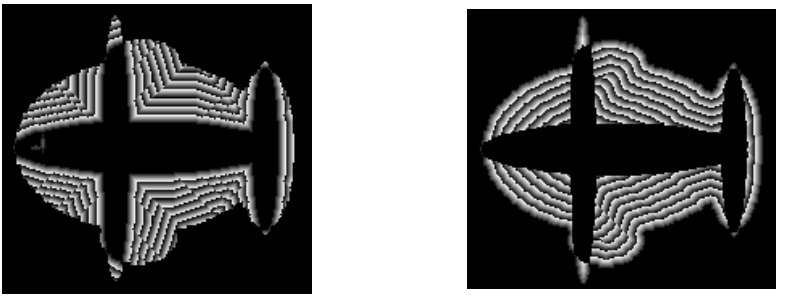

Fig. 6. The two geodesic waves: d1 (left) and d2(right)

Consequently, to each pixel of the difference we associate two distances d_1 and d_2 that are the basis of the morphing.

Erosion and Dilation. Let us denote by β a parameter, varying between 0 to 1, that gives the degree of progression of the morphing. The value $\beta = 0$ gives shape a, $\beta = 1$ gives shape b and any other value between 0 and 1 gives an intermediate shape between a and b. To obtain exactly the median image, the parameter β must be equal to $\frac{1}{2}$.

The erosion and dilation process is applied to each connected component of the difference. Some parts will grow and other will thin. In order to make proportional growing and thinning on all the connected components, we compute for the connected component labelled i its maximal distance d_{1i} and d_{2i}.

Then the decision of erosion or dilation is taken as follows. We start with a shape c initialized with pixels of shape a. A pixel of $a \backslash b$ in the i^{th} component of $a\Delta b$ is removed from c if it is labeled with distances d_1 and d_2 verifying:

$$d_1 \geq d_{1i}(1 - \beta) \text{ or } d_2 \leq d_{2i}\beta$$

A pixel of $b \backslash a$ in the i^{th} component of $a\Delta b$ is added to c if it is labeled with distances d_1 and d_2 verifying:

$$d_1 \leq d_{1i}\beta \text{ and } d_2 \geq d_{2i}(1 - \beta)$$

Figure 7 shows the different intermediate images for our example corresponding to β equals to $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$ and $\frac{1}{5}$.

The dilation and erosion step is the final step to generate the average shape. It can also be used to generate arbitrary intermediate shapes between any two shapes.

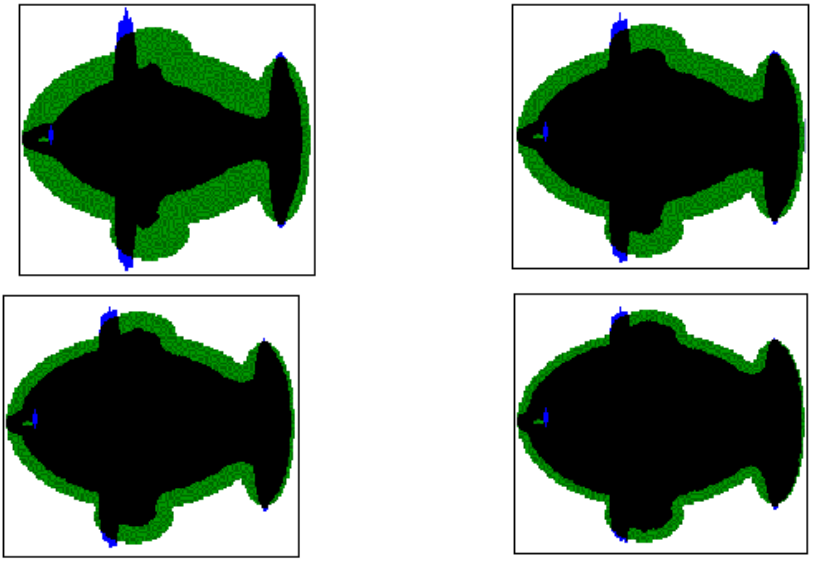

Fig. 7. Examples of intermediate images with $\beta = \frac{1}{2}$, $\beta = \frac{1}{3}$, $\beta = \frac{1}{4}$ and $\beta = \frac{1}{5}$

Table 1. Pseudo-code of the morphing algorithm

---

a=Data_file(1)
b=Data_File(2)
a and b are the two input objects
(a,b)=Registration(a,b)
Component_labeling($a \Delta b$)
/ each 4-component of* $a \Delta b$ *has its own label i */*
Geodesic_distance_transform($a \Delta b$, background)
Geodesic_distance_transform($a \Delta b$, $a \cap b$)
β=morphing_parameter
$c = a$
For each connected component i of ($a \Delta b$)
  d_{1i}=compute_maximum(d1 in i)
  d_{2i}=compute_maximum(d2 in i)
  for each pixel p in component i
    if($p \in a \backslash b$) */* Erosion */*
      if($d_1[p] \geq d_{1i}(1 - \beta)$ or $d_2[p] \leq d_{2i}\beta$)
        $c = c \backslash \{p\}$
    else */* (*$p \in b \backslash a$*): Dilation */*
      if($d_1[p] \leq d_{1i}\beta$ and $d_2[p] \geq d_{2i}(1 - \beta)$)
        $c = c \cup \{p\}$
return c

4.3 Pseudo-Code and Complexity

Table 1 is a pseudo-code Summarizing all the steps needed to deform one discrete object to another one: the complexity of our method is $O(n)$ where n is the number of pixels of the images (assumed to be of the same order of magnitude in the input and the output images). The distance information we manipulate cannot be computed with the traditional distance transform algorithms [Blu67] that are not adapted for geodesic distance transformations since we work on non-convex domains. We use an adapted version of Dijkstra Graphsearch algorithm using a priority queue indexed by the distance. Since the maximal possible distance value is at worst proportional to the size of the input, we can use a bucket data structure [CLRS01] with all points at distance i stored in a chained list of index i.

Thus, each new pixel will be stored in a chained list of index i witch is its distance value. As we can update pixels with other distance values due to new waves of geodesic propagation, it is logical to follow a rule:

- If the new distance value is bigger than the original one: we do nothing
- If it is smaller, we update the pixel distance value by deleting it from his original list and adding it in the appropriate chained list, indexed with the new distance value, at the last position.
- If for a given distance value there are no elements, we move to the next chained list indexed with the next distance value.

So, at each step we are not obliged to sort values (find the pixels with smallest values), we can obtain them by selecting elements from first chained list associated to the smallest distance value.

This structure allows us to insert a new element and to select the smallest element in constant time. A given pixel can be inserted in the bucket a constant number of times only, leading to a total coast proportional to the number of pixels of the images. In the following section we give some examples that illustrates the flexibility of our approach.

5 Results

In order to validate our method, we have made tests between discrete objects of different natures. This indicates that the approach can be very adaptable to various shapes. In figure 8 we try to find an average shape between a circular object and a fish. Another example is in figure 9 and computes an average shape between a rectangular object and a fish.

It can be noted that it is equivalent to transform the object a into object b using $\beta = \beta_0$ than to transform object b into object a using $\beta = 1 - \beta_0$. The extension to 3D is straightforward and does not need any additional operation. We use the 3x3x3 neighborhood and the 3-4-5 chamfer distance.

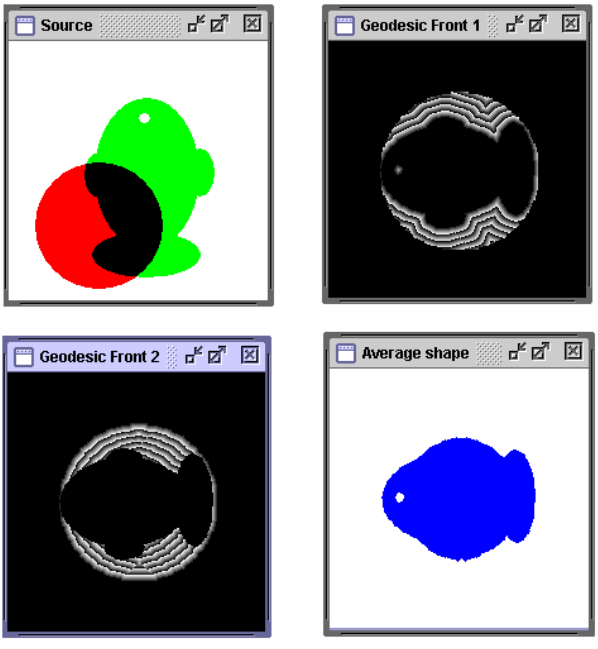

Fig. 8. Deformation of various shapes with β=1/2

6 Conclusion and Future Works

We have presented a linear algorithm for computing an intermediate shape between two arbitrary input binary shapes. A progression parameter β ranging from 0 to 1 allows to control the influence of each input shape. A generalization of this problem to n shapes could be achieved by recursively computing an average between the first $(n-1)$ ones then computing an intermediate between this new shape and the n^{th} one, using $\beta = \frac{1}{n}$. It would be interesting to study the influence, on the final shape, of the order in which these shapes are processed.

In this paper we have used the chamfer distance as a good integer approximation of Euclidian distance. An improvement in precision could be the use of the Euclidian distance itself which can also be computed in linear time as shown in [CMT04] at least for 2D domains. To our knowledge, there exist no linear algorithm for computing Euclidian geodesic distance transformation in 3D. The chamfer distance is thus probably a good choice in 3D if performance is important.

In all the examples we have shown in this paper, the topology of intermediate shapes is preserved if the two input objects have the same topology. In some cases, however, if the objects are too different it can happen that the topology of intermediate objects is not preserved (holes or disconnected components might temporarily appear). If the two objects have the same topology and their intersection after the registration step also has the same topology, we could expect to

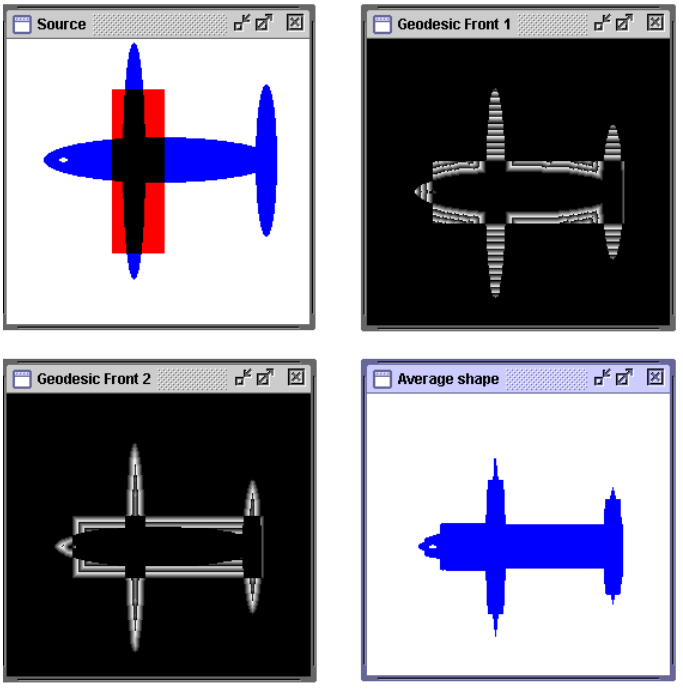

Fig. 9. Deformation of various shapes with β=1/2

find a transformation whose intermediate shapes also have the same properties. A heuristic for reaching this goal could be to use thinning or thickening operators, allowing to ensure that the removal or adding of a point [Ber96] does not change the topology of intermediate objects. These topics should be the subject of further studies.

References

[Ber96] G. Bertrand. A Boolean Characterization of Three-dimensional Simple Points. *Pattern Recognition Letters*, 17:115–124, 1996.

[Blu67] Blum. A Transformation for Extracting new Descriptors of Shape. *In models for perception of speech and visual form*, pages 362–380, 1967.

[Bor88] G. Borgefors. Hierarchical Chamfer Matching: a Parametric Edge Matching Algorithm . *IEEE transactions on pattern analysis and machine intelligence*, 1988.

[BTSG00] R. Blanding, G. Turkiyyah, D. Storti, and M. Ganter. Skeleton-based Three Dimensional Geometric Morphing. *Computational Geometry*, 15:129–148, 2000.

[CLRS01] Thomas H. Cormen, Charles H. Leiserson, Ronald L. Rivest, and C. Stein. *Inroduction à l'Algorithmique*. xDunod, Paris, 2001.

[CMT04] D. Coeurjolly, S. Miguet, and L. Tougne. 2D and 3D Visibility in Discrete Geometry: An Application to Discrete Geodesic Paths. *Pattern Recognition Letters*, 25:561–570, 2004.

[FLD02] M. Fleute, S. Lavallé, and L. Desbat. Integrated approach for matching statistical shape models with intra-operative 2D and 3D data. In *MICCAI 2002*, volume part II, pages 364–372, Springer 2002.

[Iwa02] M. Iwanowski. Image Morphing Based on Morphological Interpolation Combined with Linear Filtering, 2002. Available at: `http://wscg.zcu.cz/wscg2002/Papers_2002/A23.pdf`.

[LB98] A. Liu and E. Bullitt. 3D/2D Registration via Skeletal near Projective Invariance in Tubular Objectives. In *MICCAI*, pages 952–963, 1998.

[MMF98] C.R. Maurer, R.J. Maciunas, and J.M. Fitzpatrick. Registration of Head CT Images to Physical Space Using Multiple Geometrical Features. In *Proc. SPIE Medical Imaging 98*, volume 3338, pages 72–80, San diego CA, February 1998.

[RMP+98] A. Roche, G. Malandain, X. Pennec, P. Cathier, and N. Ayache. Multimodal Image Registration by Maximisation of the Correlation Ratio. Technical Report 3378, INRIA, 1998. Available at : `http://www.inria.fr/rrrt/rr-3378.html`.

[RP68] A. Rosenfeld and J.L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[Wol90] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, Los Alamitos, CA, 1990.

[WV96] A. Wells and P. Viola. Multimodal Volume Registration by Maximization of Mutual Information. *Medical Image Analysis*, 1:32–52, 1996.