SHAPE BASED ALIGNMENT OF 3-D VOLUME SLICES

Stelios Krinidis, Nikos Nikolaidis and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, 54006 Thessaloniki, Greece
e-mail: {skrinidi,nikolaid,pitas}@zeus.csd.auth.gr

ABSTRACT

This paper presents a method for the accurate and computationally efficient alignment of 2-D slices that form a 3-D volume. Alignment incorporates object shape information and is achieved by evaluating the forces acting upon pairs of slices whose object outline points are connected by a system of springs under tension. Experimental results on medical and synthetic 3-D images prove the efficiency of the method.

1. INTRODUCTION

3-D images (3-D volumes) are encountered in a variety of scientific disciplines like medical imaging, geophysics, meteorology etc. In medicine 3-D data are acquired by a multitude of imaging devices (MRI, CT, 3-D microscopy etc.). In most cases 3-D images are represented as a sequence of 2-D parallel image slices. Depending on the acquisition method the 2-D slices can be aligned or not. MRI images, for example, are aligned whereas slices obtained through physical sectioning, e.g. biological tissue slices obtained using a microtome, are not. In the latter case automatic or manual alignment of the misaligned slices is of utmost importance for the correct 3-D visualization and morphometric analysis (e.g. surface and volume measurements) of the objects under study. Usually alignment of such images involves only rotation and translation compensation. Image alignment is also very important for multi-sensor imaging. In this case alignment can imply correction of non-rigid affine transforms.

A number of algorithms have been proposed so far for automatic image alignment [1], [2], [3], [4], [5]. The algorithm proposed in this paper aligns pairs (or triplets) of slices using object contour information. Points on the object contour on one slice are considered as being attracted by the contour points on the other slice. The magnitude of the attraction forces is considered proportional to the squared Euclidean distance between points. The physical equivalent of this is to consider a system of springs that connect each contour point on one slice with all contour points on the next slice.

The paper outline is the following: The algorithm is presented and explained in Section 2. Simulation results on a number of 3-D images are reported in Section 3. Conclusions are drawn in the last section.

This work has been supported by the research project 99ED 599 (PENED 99) funded by GSRT and the European Social Fund.

2. ALGORITHM DESCRIPTION

Our aim is automatic alignment of a sequence of slices forming up a 3-D image. Only translation and rotation corrections are considered. As mentioned above, the method is based on object shape information which is provided to the algorithm in the form of the object outline. Therefore, a preprocessing step of manual or automatic delineation of the objects of interest is required. This can be performed for example by segmenting the objects of interest and then applying an edge detector on the segmented volume. Applying edge detection techniques directly on the grayscale images is also possible but in such a case one should take into account the erroneously detected edges which will deteriorate the algorithm performance. In its first stage the algorithm tries to align the images in successive pairs; slice 2 is being aligned to slice 1, slice 3 is then aligned to slice 2 and so on. The term *reference image* will be used from now on to denote the image with the lower index value (image k) whereas the image with the upper index value (image $k+1$) will be called *current image*. The algorithm keeps the reference image fixed and tries to rotate/translate the current image so as to become aligned with the reference image. In order to do so an analogy between the situation at hand and the physical model of two planar objects whose outline

0-7803-6542-9/00/$10.00 © 2000 IEEE

points are connected with springs of a fixed length is drawn. Each point on one contour is connected to all points of the other contour. The magnitude of the force acting upon a pair of points is proportional to their squared Euclidean distance. These forces move one object relative to the other until a minimum energy position (equilibrium) is reached. This procedure is then repeated for the next pair of slices. The algorithm terminates when all slices have been aligned.

The force $\mathbf{F_{i,j}}$ acting upon point i of the reference image as a result of it being attracted by point j on the current image (figure 1) is given by the following formula:

$$\mathbf{F_{i,j}} = K * |\mathbf{d_{i,j}}| * \mathbf{d_{i,j}} \tag{1}$$

where K is a constant (spring constant) and $\mathbf{d_{i,j}}$ the distance vector between the two points. Springs of zero initial length have been assumed in this formula.

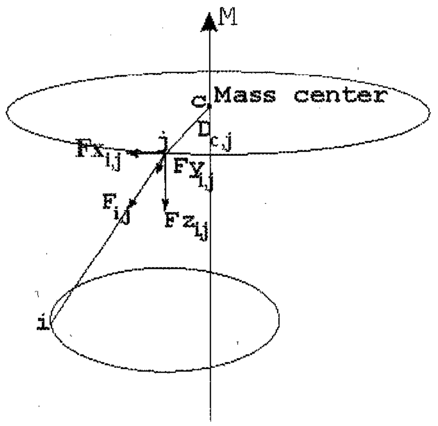

Figure 1: Example of the force acting upon two contour points

$\mathbf{F_{i,j}}$ can be analyzed to three components $Fx_{i,j}$, $Fy_{i,j}$, $Fz_{i,j}$, along the X, Y, Z axes correspondingly:

$$Fx_{i,j} = |\mathbf{F_{i,j}}| * cos(\theta) * cos(\phi) \tag{2}$$

$$Fy_{i,j} = |\mathbf{F_{i,j}}| * cos(\theta) * sin(\phi) \tag{3}$$

$$Fz_{i,j} = |\mathbf{F_{i,j}}| * sin(\theta) \tag{4}$$

where θ is the elevation angle and ϕ is the azimuth angle in a spherical coordinate system. Since the aim of the algorithm is to align the two slices while keeping their vertical distance fixed, component $Fz_{i,j}$ will be neglected in the subsequent analysis.

The torque $\mathbf{M_{i,j}}$ with respect to the object center of mass c is given by the following formula:

$$\mathbf{M}_{i,j} = \mathbf{F}_{i,j} \times \mathbf{D}_{j,c} \tag{5}$$

where $\times$ denotes vector outer product and $\mathbf{D}_{j,c}$ is the distance vector between pixel j and the object's center of mass c. Torque is a vector that is perpendicular to the image plane. If its direction is positive then the object is rotated anti-clockwise whereas a torque of negative direction results in clockwise rotation.

If the number of pixels in the contours of the reference and the current frame are N_1 and N_2 respectively, the total number of forces that act upon the points of the current image is $N = N_1 * N_2$. The components F_x, F_y of the resultant force $\mathbf{F}$ can be stated as follows:

$$Fx = Fx_{1,1} + Fx_{1,2} + \ldots + Fx_{N1,N2} \tag{6}$$

$$Fy = Fy_{1,1} + Fy_{1,2} + \ldots + Fy_{N1,N2} \tag{7}$$

The resultant torque with respect to the object's center of mass c is given by the following equation:

$$\mathbf{M} = \mathbf{M_{1,1}} + \mathbf{M_{1,2}} + \ldots + \mathbf{M_{N1,N2}} \tag{8}$$

After force/torque calculation the algorithm proceeds in translating/rotating the current slice. The translation/rotation directions are determined by the directions of the resultant force and torque. The amount of translation (in pixels) and rotation (in degrees) is taken to be proportional to the magnitude of the resultant force and torque respectively. The procedure is repeated until the resultant force and torque become zero, i.e., until the system reaches its equilibrium.

In order to improve the results, a second refining step can be performed. In this step slices are considered in triplets, slices $k-1, k+1$ being the reference slices and slice k being the current slice, i.e. the slice that is to be aligned with the other two. Springs that connect each point of the current slice with all points of the reference slices are considered in this case.

In order to minimize the number of iterations required to align a pair of slices, a preprocessing step that translates the slices so that their centers of mass become aligned can be used.

It should be noted here that instead of using only points on the object outline, the algorithm can be applied on all the points that lie on an object's interior or on a randomly selected subset of these points. However experiments have proven that incorporating interior points in the alignment procedure does not improve significantly the performance of the algorithm, especially in comparison to the corresponding increases in the computational complexity.

3. SIMULATION RESULTS

Experimentation has been carried out on synthetic 3-D images depicting geometrical shapes as well as on CT scans of mechanical parts, and medical 3-D images, i.e., sequences of parallel cross sections of teeth, viewed with an optical microscope. In the latter case the acquired slices are not aligned and thus a skilled dentist was used to manually align the slices in the first stage. Then the slices were randomly rotated and translated and fed to the algorithm for alignment. The same procedure was applied on the synthetic and CT images but this time no manual alignment was performed since these images were already aligned.The following performance measures have been used for judging the efficiency of the algorithm:

	Translation Error (pixels)	Rotation Error (degrees)
Synthetic 3-D Image (ellipsoid)	0.22	0.77
CT scan (engine)	0.16	1.01
Tooth 3-D Image	0.35	3.93

Table 1: Translation and rotation errors

- Translation Error: the mean translation error (in pixels) averaged for the X, Y directions. The error is evaluated by finding the translation between the algorithm-aligned slice and the actually aligned slice, for all pairs of slices.

- Rotation Error: the mean rotation error (in degrees). The error is evaluated by finding the rotation between the algorithm-aligned slice and the actually aligned slice, for all pairs of slices.

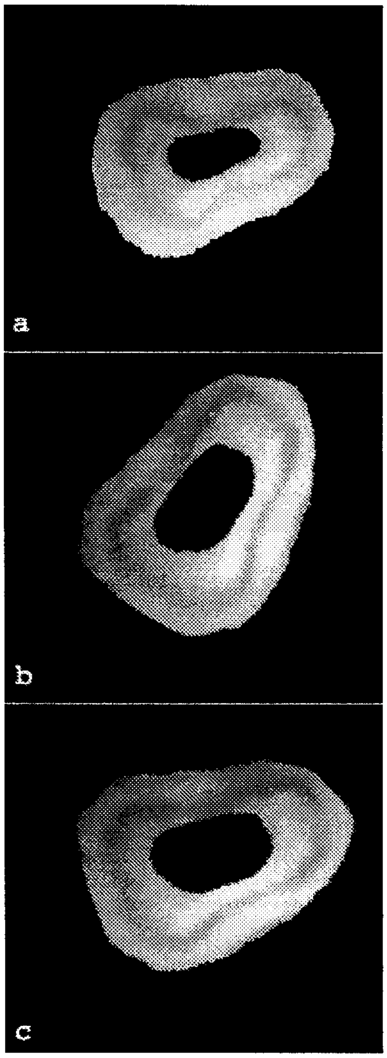

Figure 2: Alignment results on two consecutive slices from a tooth 3-D image. (a) Reference image; (b) current image prior to alignment; (c) current image after alignment

Results are presented in Table 1 for two synthetic and a medical 3D image.

Alignment results are visualized in figures 2 - 4 where slices from a tooth 3-D sequence, a synthetic 3-D image of an ellipsoid and a CT scan of an engine block are depicted. It is obvious that the two slices are efficiently aligned. Similar results were obtained when the dissimilarity between the two slices was more pronounced.

4. CONCLUSIONS

An efficient automatic alignment algorithm has been proposed in this paper. The algorithm performs satisfactorily in both synthetic and real-world 3-D im-

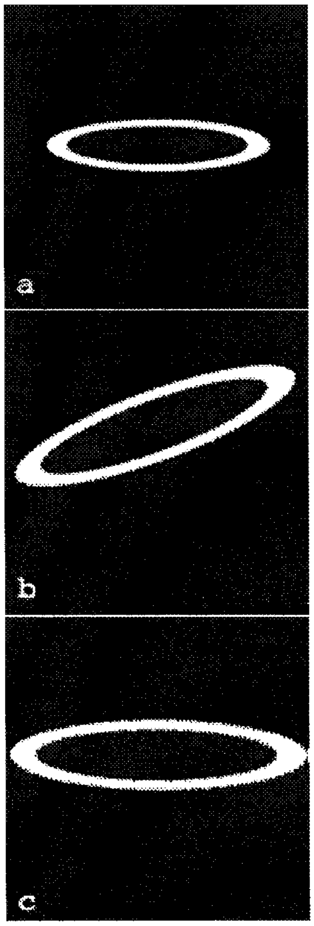

Figure 3: Alignment results on two consecutive slices from a synthetic 3-D image depicting an ellipsoid. (a) Reference image; (b) current image prior to alignment; (c) current image after alignment

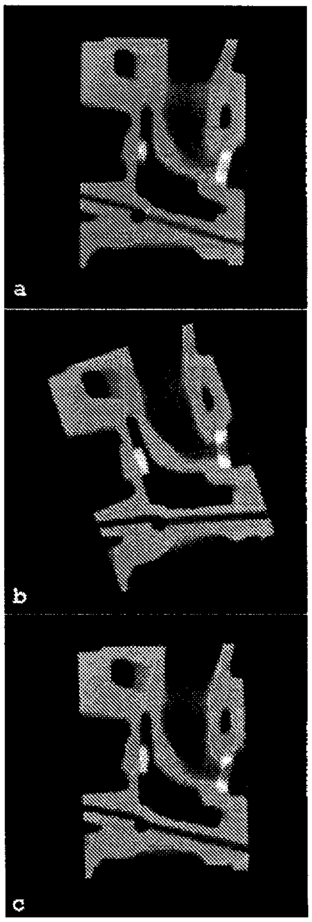

Figure 4: Alignment results on two consecutive slices from a CT scan of an engine block. (a) Reference image; (b) current image prior to alignment; (c) current image after alignment

ages. In the near future an extension of the algorithm that will combine shape information with image intensity information will be pursued.

5. REFERENCES

[1] P. Viola and W. Wells, Alignment by Maximization of Mutual Information, in *International Conference on Computer Vision*, pages 16-23, Cambridge, MA, June 1995.

[2] M. Irani and P. Anandan, Robust Multi-Sensor Image Alignment, in *IEEE International Conference on Computer Vision (ICCV)*, India, January 1998.

[3] K. J. Dana and P. Anandan, Registration of Visible and Infrared Images, in *Proc. SPIE Conf. on Arch., Hardware and FLIR in Auto. Targ. Rec.*, pages 1-12, 1993.

[4] H. Li, B.S. Manjunath and S. K. Mitra, A Contour-based Approach to Multi-sensor Image Registration, *IEEE Trans. on Image Processing*, pages 320-334, 1994.

[5] R. Kumar, K. Dana and P. Anandan, Frameless Registration of MR and CT 3D Volumetric Data Sets, in *Proc. of the Workshop on Applications of Computer Vision II*, 1994.