

Multiple Surfaces Reconstruction from 2D Sections Using an Increasing 2D Vector Flow

Nikolay Metodiev Sirakov
Dept. of Mathematics, Dept. of Computer Science
Texas A&M University Commerce
Commerce, TX 75 429

Abstract: *This paper presents a new approach to automatically construct multiple surfaces from set of 2D sections. A convex curve and centripetal normal force are employed to split each section to a set of shells. Each shell contains a single image region and defines an initial contour evolved by the geometric heat differential equation in the direction of a centripetal force toward the outer boundary of the image region. Then a re-parameterization is performed to increase the flow and make each contour converging into concavities. Thus, the 2D sections are segmented to a set of contours, divided to subsets of similar contours. Each subset is used to construct the surface of a single 3D object linking corresponding vertices. To validate the theory a set of experiments is performed using synthetic and medical 2D sections. A discussion and comparison of the method with set of existing is given at the end of the paper.*

Key Words: normal force, 2D segmentation, corresponding vertices, concavities, 3D visualization.

1.0 Introduction

The automatic 3D objects reconstruction and visualization is subject of great interest for the Medical and Bio-medical imaging [1,2,3,9], because the smooth and correct 3D visualization of an organ opens a new dimension for its anatomic study and analysis. To reach the goal a large amount of data has to be processed, and it calls for efficient and low-cost numerical methods and algorithms.

Two kinds of input data are used in the field: volumetric; and 2D images. Methods designed to use the latter data include visualization by multiple view points, shape from shading [6], and surface generation connecting points on adjacent 2D sections. For instance [3,9] use a set of non parallel 2D sections, apply Delaunay triangulations, and Voronoi diagrams to build the surface. In [13] smooth functions are used to design the 3D shape of the objects. A 3D reconstruction and visualization approach, and tool based on essential points, and regularities are reported in [14].

During the past decade, PDE-driven contour and surface deformations [1,10,11,19,21,22] have become very popular in the computer vision community for shape recovery and object detection. In general, deformable models [20] can be divided into two categories: explicit and implicit models.

Explicit models include parametric [24] and discrete representations [18]. In [8] partial differential equation evolves an active surface to construct the shape of bio-medical structures. A model based on the heat differential equation and vector flow is reported in [15]. Using a centripetal vector flow, the method segments a 3D image/scene to multiple 3D regions (shells) and reconstructs their shape [15].

Most of the existing implicit algorithms are based on the Eulerian approach, i.e., the geometry and topology of the shape is implicitly defined as the level-set solution of time-varying implicit functions over the entire 3D space [5]. Implicit models [5,7,20,23] handle topology changes, based on the modeling of propagating fronts, which are the level set of some scalar function. A disadvantage of these models is their high computationally cost. To decrease it a fast marching-cube technique was developed by Sethian and Malladi [12]. Despite of the theoretical potential of the implicit models to handle splitting and merging surfaces there is a lack of experimental results, in the literature, regarding automatic 3D image/scene segmentation and multiple 3D objects reconstruction.

This paper presents an approach whose input is a set of 2D sections (images-which could be DICOM) or set of 3D points. In the latter case a preprocessing is made to project the points onto a set of planes. In every 2D section the method runs an explicit active contour with shells [16,17] and

increasing, inward vector flow. Shells Algorithm (SA) uses the flow and a penalty function to split each 2D section to set of distinct conic regions - shells [16,17]. The subdivision continues until a single region remains in every shell, which defines an initial active contour with an inward normal vector flow, guided by the geometric heat equation to the boundary of the region. The dimension of the vector flow is selected to be one and the same for every active contour in each 2D section. Once the boundary of every image region is determined a recognition tool is employed to divide the entire set of boundaries to sub-sets of similar boundaries [14]. Each subset is used for surface construction of a single 3D object by linking corresponding boundary vertices which lie in adjacent 2D sections.

The rest of the paper is organized as follow: section 2 presents SA, section 3 shows the 2D active contour model, 4 describes its discrete implementation. In section 5 the vertices from a subset of similar contours are tiled to a rectangular surface of a single object. Two different cases are considered for surface reconstruction: the entire boundary is visible by a centripetal force; and otherwise. A set of experiments is performed to validate the theory. Medical and synthetic images of different modalities and varying sizes are used for this purpose. The advantages and contributions of this work are listed at the end of this paper, where our method is compared with others in the field.

2.0 Shells algorithm for image segmentation

Consider an initial smooth-convex curve $C(t,s)=r(t,s)=x(t,s)i+y(t,s)j$ which envelops the entire 2D section, where $t \in [0,\infty)$ parameterizes the family of curves, whereas s is an arc segment, which parameterizes the particular curve. Using $C(t,s)$ a centripetal normal force \vec{N} is defined and extended to the center of the 2D section [17]. A penalty function (PF) is designed to determines the rate of change of the 2D section's function $I(x,y)$ on an interval $\Delta_t \in \vec{N}$, with respect to decreasing arc length s along the normal vector \vec{N} at the point $r(t,s)$ on the curve C at time t [17]. The discrete form of the PF is defined as follow:

$$P_{\varepsilon,\varepsilon_1} = P_i^j(\Delta_t) = \begin{cases} 0 & \text{if } \varepsilon_1 \geq PS_i^j(\Delta_t) \geq \varepsilon \\ 1, & \text{if otherwise} \end{cases} \quad (1)$$

where $PS_i^j(\Delta_t) = \sum_{u=w+1}^{w+\Delta_t} |I_i^j(u) - I_i^j(u-1)|$ for $w=0,1,2,\dots,|N_i|-\Delta_t$, and $I_i^j(u)$ denotes the u -th pixel, which lies on the i -th normal vector at time j .

If, for a normal vector \vec{N} , there exists $\Delta_t \in \vec{N}$ such that $P_{\varepsilon,\varepsilon_1}(\Delta_t) = 0$ we say that \vec{N} is a zero vector. Otherwise we call \vec{N} a nonzero vector. If all vectors in a force are zero we say that the force is zero. Otherwise the force is non zero. A number of consecutive zero vectors we denote by Δ_s and call it a radial size of an image region.

The thresholds $\varepsilon, \varepsilon_1, \Delta_t, \Delta_s$ are assigned by the user to describe the smallest image region subject of interest. The SA ends for a normal force, where all vectors are zero vectors (this we use as a stop criterion). Otherwise, the algorithm determines the shells as conic segments determined by consecutive zero vectors [16, 17]. Thus the normal force splits a 2D section, say the m -th one, to shells Sh_{km} , $k=1,2,\dots$. In each shell Sh_{km} the algorithm computes the mass center $\bar{\mu}_{km}$ and computes the maximal distance R_{km} from $\bar{\mu}_{km}$ to the boundaries of Sh_{km} . Both R_{km} and $\bar{\mu}_{km}$ are used as radius and center of initial active contour C_{km} inscribing Sh_{km} . Further a centripetal normal force, with one and same size is defined for each C_{km} and the PF is computed for each vector in the force. If the normal force is zero in Sh_{km} the stop criterion is satisfied and the process of subdivision ends for this shell, which holds a single region, containing the center of Sh_{km} . If the force is non zero in Sh_{km} the process of subdivision continues, as described above.

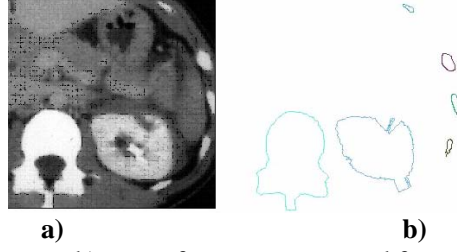


Figure 1: a) A 2D section of torso; b) a set of organs segmented from this section.

3.0 Active contour with increasing flow

Recall that at the end of the subdivision process a single curve C_{km} is determined for each Sh_{km} . The

curve C_{km} is evolved, to the boundary of the region inscribed in the shell, by
$$\frac{\partial C_{km}}{\partial t} = P \frac{d \vec{\tau}}{ds}, \quad (2)$$

where $\vec{\tau}(s) = \frac{x_s 2}{nc} i + \frac{y_s 2}{nr} j$ denotes the tangent vector normalized by the image's sizes. By nc and nr we denote the 2D section's number of columns and rows respectively; $P = P_{\varepsilon, \varepsilon_1}(\Delta_t)$ is defined by Eq.1. Note that Eq.2 represents a normal force which ends the evolution of C_{km} if the force becomes zero. At this stage a re-parameterization of every active contour C_{km} is performed using the minimal arc length segment s . Then a normal vector is computed at every newly added point and C_{km} is evolved in the direction of the normal force. Thus, the active contour's segments which coincide with boundary portions will stay motionless, whereas some segments with newly computed vectors will converge to a concavity boundary (see Figs2a) and 5b)). After that a new re-parameterization is performed and again inward normal vector is computed at each newly added point, and every curve C_{km} is evolved by Eq.2 in the direction of the normal vectors. The process of re-parameterization and evolution ends if the stop criterion is met and the length of the longest normal vector becomes smaller than Δ_t , which represents the error of boundary approximation.

The main contribution of the re-parameterization technique is to increase the force dimension and helps the active contour quickly converge into deep, spiral concavities and splits contours (Fig.5b)).

4.0 The Discrete Active Contour Model

Recall that SA splits an image to shells using the set $\{\varepsilon, \varepsilon_1, \Delta_t, \Delta_s\}$ together with the PF defined by Eq.1. Note that the set of thresholds and the dimension of the normal force n are one and the same for each initial C_{km} , $m = 1, \dots, \mu$, and $k = 1, \dots$. Denote by $r_{ikm}^j = (r^j(t^j, s_i^j))_{km}$ for $i=1, \dots, n$, and $r_0^j = r_n^j$ the vector function which defines C_{km} - k -th initial contour in the m -th 2D section. For purpose of simplicity we omit for a while the indexes m and k kipping in mind we work in the m -th section with the k -th contour. By j we parameterize the family of curves, while i parameterizes the particular curve; $s_i^j = |r_i^j - r_{i-1}^j|$ is the i -th discrete space step at time j ; $\partial \vec{\tau}_i^j \approx (\vec{\tau}_i^j - \vec{\tau}_{i-1}^j) / s_i^j$; and δ_i^j is a discrete time step (from time derivative) such that $\delta_i^j = \delta^j$, for $i = 1, \dots, n$. Thus we rewrite Eq.2 in the form of discrete

active contour forward difference algorithm (ACFDA):
$$r_i^{j+1} = r_i^j + \delta_i^j \partial \vec{\tau}_i^j P_i^j. \quad (3)$$

According to [4], the ACFDA defined by Eq.3 will stably converge to the solution of Eq.2 if and only

if the time and space steps (δ^j and $\Delta s^j = \sum_{i=1}^n s_i^j / n$) are related by:
$$\delta^j < \left(\frac{1}{n} \sum_{i=1}^n s_i^j\right)^2 / 2c. \quad (4)$$

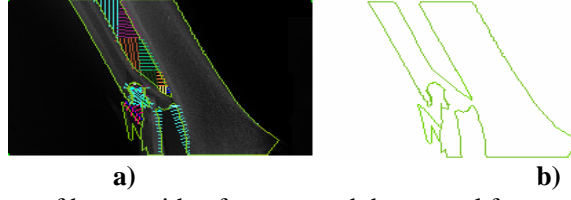


Figure 2: a) X-ray image of bones with a fracture, and the normal force converging between them; b) the contour of the bones and the fracture.

5.0 The 3D Model

In our work we consider the 3D model of an object as a collection of 2D similar boundary contours [14], obtained by the active contour model, with shells and increasing flow. Every boundary is approximated by set of vertices labeled by the following sequence: $r_{1km}^t, r_{2km}^t, \dots, r_{nkm}^t$, (5)

where r_{ikm}^t denotes the i -th vertex on the k -th contour, located in the m -th 2D section at time t .

The surface mesh is generated as a collection of spatial rectangles drawn by connecting corresponding contour vertices: r_{ikm}^t and r_{jw}^t . Two vertices are called corresponding if they belong to similar contours - $k=v$, which lie on consecutive 2D sections - $w=m+1$ and their labels (consecutive numbers) on the contour are equal- $i=j$. To receive a triangular mesh every spatial rectangle is divided to triangles using the principle of the nearest neighbor.

5.1 Reconstruction in case of visible boundaries

Consider a set of 2D image regions cut from a 3D object. Assume that the entire boundary of every image region is “visible” by the corresponding vector flow, for example the brain boundaries in Fig.3. In this case a re-parameterization is not needed, and will not be performed in the process of evolution. Recall that a force with the same dimension is determined for every initial active contour. Therefore, when a convergence is accomplished every boundary will be approximated by one and the same number of vertices, and every vertex will have a corresponding one on a similar boundary located in a neighbor 2D section. Follows that the surface between every pair of similar contours is build up by linking the corresponding vertices as is given below:

$$(r_{1km}^t r_{1km+1}^t), (r_{2km}^t r_{2km+1}^t), \dots, (r_{nkm}^t r_{nkm+1}^t). \quad (6)$$

In the above expression $(r_{ikm}^t r_{ikm+1}^t)$ denotes the spatial edge which links the i -th points on the k -th contours (boundary) located in the m -th and $(m+1)$ -st 2D sections. From general point of view the number of time iterations taken by an active contour to approximate different boundaries is different: $t_m \neq t_{m+1}$. Eq.6 represents the surface between pair of similar boundaries in terms of spatial rectangles. Each of them is divided to triangles by using the principle of the nearest neighbor.

The above algorithm was applied to construct the surface defined by the set of brain sections given in Fig3a). Each 2D section is of size 178x228. The reconstructed brain portion is visualized in Fig3b).

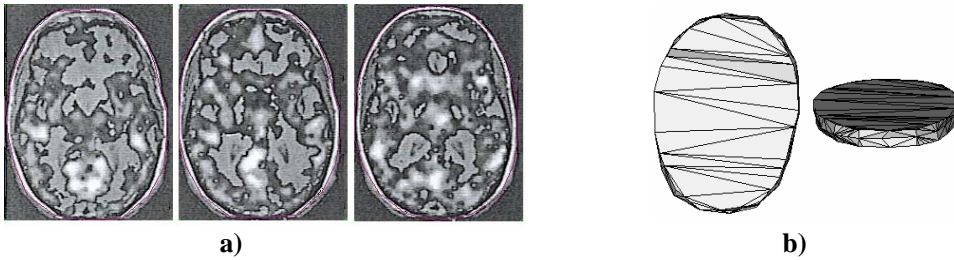


Figure 3: a) Three MRI brain sections whose boundary was determined by ACFDA. b) 3D portion of a brain reconstructed using the sections from a). Top view and view after rotation to 75 deg.



Figure 4: **a)** 3D reconstruction of the organs, from Fig2b), by using two sections; **b)** a view after rotation by 60° about Ox axis and 30° about Oy axis.

The boundary of each brain regions is extracted in: 0.15, 0.20, and 0.16 sec respectively. The surface was constructed and painted in 0.5 sec using a PC with CPU 1.8GHz. A vector flow with 20 normal vectors and $\Delta_r = 2$ pixels were used to perform the experiments.

Consider two sections as those shown in Fig.1a). Employing the algorithm given above each of these sections was automatically segmented to six image regions. The entire set of contours was divided to six subsets of similar contours (boundaries) by employing similarity measuring tool given in [14]. Each subset was used to construct the surface of a single 3D object (Fig4a). The six surfaces were generate in 5.5 sec using the above mentioned computer platform. The size of each 2D section is 283x290 pixels.

5.2 Reconstruction in case of partially visible boundaries

Consider two image regions whose entire boundary can not be observed by the corresponding vector flow. That is the case of deep or spiral concavities (Fig5a). Assume that the outer boundaries of the two regions belong to the k -th subset of similar contours, and are located in two adjacent sections, m and $m+1$ (Fig5a)). Denote by $r_{ikm}^{t_m} r_{(i+1)km}^{t_m}$ the edge, which closes the i -th concavity of the k -th contour in the m -th section at time t_m and call it a concavity edge. The subscript i shows also the position of the concavity edge in the sequence of edges approximating the outer contour. Analogously $r_{jk(m+1)}^{t_{m+1}} r_{(j+1)k(m+1)}^{t_{m+1}}$ is the edge, which closes the j -th concavity of the k -th contour on the $(m+1)$ -th section at time t_{m+1} . Again j shows the position of the concavity edge in the sequence of vertices approximating the outer contour (Fig5a). We will say that two concavities are corresponding if and only if $|i - j| \leq \Delta_s$. Otherwise the concavities are not corresponding.

Recall that when the outer (visible by the centripetal flow) boundary is determined the method re-parameterizes each C_{km} by using the minimal arc segment s in order to make the contour capable of converging into concavities. The algorithm halts this convergence if both criteria are satisfied: the force is zero; and the longest vector is smaller than Δ_r . Thus the set of vertices, which approximate the i -th concavity, is added to the set of vertices approximating the outer part of the k -th contour in the m -th section:

$$r_{ikm}^{t_m}, r_{ikm}^{t_m+1}, \dots, r_{ikm}^{t_m+p} = r_{(i+1)km}^{t_m}. \quad (7)$$

The following vertices approximate the j -th concavity of the k -th contour in the m -th section:

$$r_{(j+1)k(m+1)}^{t_{m+1}}, r_{jk(m+1)}^{t_{m+1}+1}, \dots, r_{jk(m+1)}^{t_{m+1}+q} = r_{(j+1)k(m+1)}^{t_{m+1}}. \quad (8)$$

Determine $\max(p, q)$, say p . Using the number p the method re-parameterizes the boundary of the j -th concavity on the k -th contour in section $m+1$. Thus both concavities are parameterized by one and the same number of arc segments. Follows that every vertex on a concavity's boundary has a corresponding one and the algorithm links them. Thus the surface is created by means of spatial rectangles. Since we would like to have boundary approximation by plane segments the algorithm divides every spatial rectangle to triangles. Note that in this case we consider vertices' correspondence with a displacement of $|i - j|$.

The above algorithm was employed to generate a surface using the contours of the regions shown in

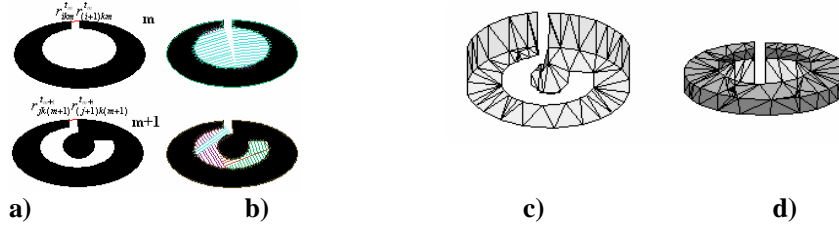


Figure 5: **a)** two 2D objects with deep and spiral concavity; **b)** the normal forces used to determine concavities' boundary; **c)** 3D reconstruction using the objects from a); **d)** top view.

Fig.5a). To obtain the boundary vertices of each region we used an initial vector flow with 30 vectors and $\Delta_s = 2$. The boundary of each region was obtained in about 0.18 sec. Each image is of size 269x196. The surface was generated in 0.7 second. The used displacement to define concavity correspondence is 0, because $i=j$.

If $|i-j| > \Delta_s$, no one of the concavities has a corresponding one. In this case the algorithm divides the edge $r_{ik}^{l_{m+1}} r_{(i+1)k}^{l_{m+1}}$ to p arc segments and the edge $r_{jk}^{l_m} r_{(j+1)km}^{l_m}$ to q arc segments and provides corresponding vertex to each concavity vertex. In this case the correspondence is considered with zero displacement.

6.0 Advantages and Discussion

A contribution of this paper is the application of 2D active contours with shells and increasing flow to segment a 3D volume and visualize multiple 3D objects (see Fig.4). Such an opportunity is not provided by any of the methods reported in [1,10,11,21,24].

Recall the input to our method is a sequence of 2D sections, which do not need to be parallel, because no parallel property of the sections is used. The same input as above is used in [2], where active nets are employed to segment each section to multiple contours constructing the surface of the bones. A disadvantage of this approach, with respect to our, is the need of a prior histogram analysis to determine the distance map used by the active nets.

Another explicit deformable model whose input is a sequence of 2D sections and employs 2D active contours is developed in [11]. To generate the surface a sequence of morphological dilations is combined with a 2D propagation front. The method possesses a very good asymptotic calculation complexity in the order of $O(MxN)$, where M is the number of the 2D sections, whereas N is the number of the grid points. The complexity of the morphological dilation is less than $O(N)$.

The calculation complexity of our method is in the order of $O(jnM)$, where M is defined as above, n denotes the dimension of each flow, j is the maximal number of time iterations taken by the active contours to reach the boundary of a 2D image region.

As one may tell from above the two calculation complexity expressions use notions of different nature. Therefore a direct comparison is not possible. But one advantage of our approach over the method given in [11] is that the latter is semi-automatic, whereas our does not need user interaction.

Active surface models, which handle volumetric data are reported in [1,8,10,21,24]. Paper [8] can use also 2D images from multiple view points. But no info is provided in these papers regarding volume segmentation and shape modeling of multiple 3D objects. One may observe from Figs1 and 4, our method automatically segments a volume, visualizes multiple 3D objects, and could handle 3D objects with complicate concavities' structure (see Fig.5).

A 3D shape modeling from 2D sections using active contours to extract image regions' boundary could be performed by the Triangle Software [25], which can handle low contrast medical images. The main disadvantage of this software is that the user selects a set of input points to be used for definition of the initial active contour, which is defined automatically by our method.

7.0 Acknowledgement

Part of this research is sponsored by a research enhancement grant funded by the Graduate School - TAMUC.

8.0 References

- [1] J. Ahlberg. Active Contours in Three Dimensions. Project #LiTH-ISY-EX-1708, Computer Vision Lab., Linköping University, Sweden, 1996.
- [2] F. Ansia. et al. Automatic 3D Shape Reconstruction of Bones Using Active Nets Based Segmentation. Proc. of the 15th Intl. Conf. on Pattern Recognition, v. 1, 486-489, 2000.
- [3] J. Boissonnat, B. Geiger. Three-dimensional reconstruction of Complex shapes based on the Delaunay triangulation. In Proc Biomedical Image Processing and Biomedical Visualization, In: R. S. Acharya, D.B. Goldgof (Eds.), San Jose CA, Vol.1905, pp. 964-975., 1993.
- [4] R. Burden and J. D. Faires. Numerical Analysis, 4th Edition. PWS-KENT Publishing, US, 1993.
- [5] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.
- [6] W.P. Cheung , C.K.Loe, K.C.Li, Direct Shape from Shading with improved rate of convergence. *Journal Pattern Recognition*, Vol.30, No 3, pp.353-365, 1997.
- [7] C. A. Davatzikos and J. L. Prince. An active contour model for mapping the cortex. *IEEE Trans. Med. Imag.*, vol. 14, pp. 65–80, 1995.
- [8] Ye Duan, Liu Yang , Hong Qin, and Dimitris Samaras. Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces. *Lecture Notes in CS 3023*, pp 238 -- 251, 2004.
- [9] B. Geiger. Three Dimensional modeling of human organs and its application to diagnosis and surgical planning. Report de Recherche N^o2105, INRIA, Sophia Antipolis, France, 1994.
- [10] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi. Gradient Vector Flows and Geometric Active Contour Models. *ICCV 1995*: 810-815, 1995.
- [11] H. Li, A. Elmoataz, J. Fadili, S. Ruan, B. Romaniuk. 3d medical image segmentation approach based on multi-label front propagation. *ICIP 2004*: 2925-292, 2004.
- [12] R. Malladi, J. A. Sethian, B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, 1995.
- [13] D. Moody, S. Lozanoff, Surf driver: A practical computer program for generating 3D models of anatomical structures. Proc 14th American Ass. of Clinical Anatomies, July, Hawaii, pp.123-135, 1997.
- [14] N.M. Sirakov, F. Muge. A system for reconstructing and visualizing three-dimensional objects”, *Computers & Geosciences*, 27, (1), 59-69, 2001.
- [15] N. M. Sirakov. Heat Equation to 3D Image Segmentation. In N. Calaos, W Lesso, W-J Tsauro, S. Jivkova, O Goriachkin (Ed), *WMSCI2005* July, Orlando, USA, pp.295-299, 2005.
- [16] N.M.Sirakov, 2005, A New Active Convex Hull Model for Image Database’s Search Space Partitioning, *In H.R. Arabnia, Xiangjing He, Tom Hintz, Qiang Wu (Ed), CSREA Press-USA*, June, 2005, pp.134-140. *ISBN: 1-932415-65-3*
- [17] N.M. Sirakov, Simonelli, I., A New Automatic Concavities Extraction Model, Published by IEEE, Computer Society, 2006, USA, 178-182. *ISBN: 1-4244-0069-4*
- [18] G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *Int’l J. Comp. Vis.*, vol. 11, no. 1, pp. 25–44, 1993.
- [19] G. Sapiro. *Geometric Partial Differential Equation and Image Processing*. Cambridge University Press, 2001.
- [20] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, vol. 4, pp. 306–331, 1988.
- [21] C. Xu, J. L. Prince. Snakes, Shapes and Gradient Vector Flow. *IEEE Tran. on Image Proc*, V.7-3, pp. 359-369, 1998
- [22] G. Zamora, H. Sari-Sarraf, S. Mitra. Estimation of Orientation of Cervical Vertebrae for Segmentation with Active Shape Models”, *Proc. SPIE Medical Imaging*, San Diego, Feb. 2001.
- [23] L. Vese, T. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *Int. J of Computer Vision*, v. 50, n. 3, pp. 271-293, 2002.
- [24] R. T. Whitaker, Volumetric deformable models: active blobs, Tech. Rep. ECRC-94-25, European Computer-Industry Research Centre GmbH, 1994.
- [25] Triangle Software. <http://www.trianglsoftware.com/>, 2005.