Heat Equation to 3D Image Segmentation

Nikolay Metodiev Sirakov
Department of Mathematics, Department of Computer Science
Texas A&M University-Commerce
Commerce, TX 75429, USA

ABSTRACT

This paper presents a new approach, capable of 3D image segmentation and objects' surface reconstruction. The main advantages of the method are: large capture range; quick segmentation of a 3D scene/image to regions; multiple 3D objects reconstruction. The method uses centripetal force and penalty function to segment the entire 3D scene/image to regions containing a single 3D object. Each region is inscribed in a convex, smooth closed surface, which defines a centripetal force. Then the surface is evolved by the geometric heat differential equation toward the force's direction. The penalty function is defined to stop evolvement of those surface patches, whose normal vectors encountered object's surface. On the base of the theoretical model Forward Difference Algorithm was developed and coded by *Mathematica*. Stability convergence condition, truncation error and calculation complexity of the algorithm are determined. The obtained results, advantages and disadvantages of the method are discussed at the end of this paper.

Keywords: 3D centripetal force, multiple surfaces, reconstruction, CBIR.

1. INTRODUCTION

During the past decade there has been an enormous increase of the 3D graphical models obtained from a variety of sources: 3D scanners, graphical 3D design software, computerized tomography, multiple camera views [8] or set of cross sections [3,8,11]. At the same time, 3D models and visualization are now used routinely in the medical diagnosis, surgical planning, mine planning, robotics, graphics design and 3D entertainment/movie industry. To store and save these 3D models image databases are built. Therefore techniques for indexing and retrieving 3D graphical models efficiently and accurately are fast becoming indispensable if we wish to query a 3D graphical/image database by examples [11,15,19]. Moreover, decision support techniques such as case-based reasoning can produce a stronger need to retrieve images that can be valuable for supporting certain diagnoses [15] as well as evaluating oil or mine deposits [32]. The need for more accurate or precise retrieval leads to the usage of active learning paradigm to exploit machine learning techniques with a human in the loop to learn and extract relevant semantics from multimedia content [11]. Linking 3D visualization, interpolation/extrapolation, learning, and tracking functions to a content based image retrieval (CBIR) will help the user to better investigate the subjects of his interest and will let him reach rapid and correct decisions.

Thus, to tackle the problem for accurate retrieval of narrow set of 3D graphical data, particularly, a volumetric or polygonal mesh model is a key to provide a good environment for decision making. In the field of CBIR the main research efforts are directed toward the development of methods capable of finding suitable and robust representation of features that are able to describe and discriminate a set of 3D models (objects) [11,18,19,30]. The latter calls for development of a metric capable of quantifying suitable similarity definition among 3D models and theirs efficient 3D comparing. Follows that the primary concern for 3D models retrieval is the extraction and representation of theirs shapes and/or theirs 3D surface textures [11]. In this paper we consider the geometric heat differential equation as a 3D mesh model to be used for 3D shape description and presentation in a CBIR system with decision support abilities.

In the sequence of tasks to be achieved for automatic retrieval of 3D objects is the problem for 3D image/scene segmentation. This problem is also a subject of great interest in the Mobile Robotics, Medicine [1,2,4,9], and Mining Industry [22,32,33].

As a matter of fact, a system designed to deal with 3D segmentation should be capable of processing a large amount of data that could come from stereo camera in Robotics, computer tomograph in Medicine, boreholes or drills in Geology. The vast amount of data requires accurate and low-cost numerical methods, algorithms and software tools capable of running in a quasi real time.

To respond to the above requirement the present paper develops a method, which uses centripetal force and a penalty function to segment the entire 3D scene/image to regions containing a single 3D object. Each region is inscribed in a convex, smooth closed surface, which defines a centripetal force. Then the surface is evolved by the geometric heat differential equation toward the force's direction. The penalty function is defined to stop evolvement of those surface patches, whose normal vectors encountered object's surface.

Once the scene/image is segmented and the 3D objects are reconstructed geometric features are to be extracted and employed for matching objects [13] and/or content-based image retrieval [14,16,30].

One can find a number of different 3D models that could be appropriate for objects' representation in a CBIR, and used for reconstruction and visualization. The first in the list is DICOM. Another approach developed in mid 90 is presented in [4,9] and applies Delaunay triangulations and Voronoi diagrams. In paper [17] smooth functions are used to build 3D object surfaces. The method is successfully implemented in a software tool called Surf Driver [34]. A 3D reconstruction and visualisation approach, and tool based on essential points, and regularities are reported in [22]. Later, a 3D interpolation technique and tool based on Mathematical Morphology was incorporated into the reconstruction method [23]. A 3D iterative modelling of the subsurface involving geophysical applications such as 3D ray tracing, 3D tomography and migration, reservoir simulation of complex geological structures is reported in [21]. Its software implementation is called gOcad, and is presented in [33]. The tool offers an opportunity for multiple surfaces intersection and union. A powerful 3D model to help open mine management is offered by GemCom [32].

In the past decade applications of differential equations (DE) and variational methods (VM) in computer science experienced a significant growth, providing the area with powerful methods and tools capable of image segmentation, objects visualization and features extraction. The concept of employing DE and VM for Image Processing [27] led to a number of significant results in image filtering, interpolation, segmentation [27,31], and 3D reconstruction and visualization [1,2,8,29]. To guide curve or surface evolution some of the models use partial DE [7,8,12,17], whereas others minimize an energy functional [1,6,29,31] and apply level sets. The level set approach is a relatively new technique, which attracts the attention of a number of scientists because it allows automatic change in the topology, and several objects can be detected simultaneously without previous knowledge of them and without using special tracking [6,10,20,28,31]. The main disadvantage of the level set methods is the relatively high run-time of the software tools based on them.

The 2D and 3D active contour models are experiencing successful applications to CBIR [2,11,15,30,32]. In this field, beside the image segmentation and features extraction, they are employed in building 3D active shape or point distribution models of deformable 3D objects such as soft-tissue organs.

This paper presents an active surface model (ASM) to be used for 3D objects representation in an image database as well as for 3D image/scene segmentation and multiple objects' shape reconstruction. The model employs the geometric heat DE to guide a surface evolvement toward the inward normal vectors [7,8,12], employed to determine object's surface points which are used later to reconstruct the shape of the object as a 3D mesh. In case of multiple 3D objects, the vector force is first employed to split the 3D image/scene to conic regions, which we will call shells. The process of segmentation continues until a single 3D object remains in each shell. Then a smooth, closed, convex surface is defined to inscribe each shell and to define an inward vector force. A new penalty function is developed to calculate the rate of change of the 3D image function on each member of the vector force. Thus, the PF is used to distinguish background and foreground voxels. The latter are used together with the normal force to generate the object's surface mesh. The Forward Difference Algorithm was employed to implement the ASM. A stability convergence condition, truncation error and calculation complexity of the discrete model are determined in the paper. *Mathematica 5.0* is used to code the discrete algorithm and design active surface software (ASS). A set of experiments was performed to validate the theory. As an input ASS uses a set of 2D sections (images) cut from 3D scene by a stereo camera (in robotics), computer tomograph (in medicine), and drills or boreholes in Geology. ASS could also use a file, whose components are vectors of the form $\{x_i, y_i, z_i, gl_{Ri}, gl_{Gi}, gl_{Bi})$, where $\{x_i, y_i, z_i)$ present the voxel's coordinate, while $gl_{Ri}, gl_{Gi}, gl_{Bi}$ show the color value of the voxel, given by the R, G, B channels.

The ASM introduced in the present paper has the following advantages over the 3D active models given in [1,2,6,12,29]: 3D scene/image segmentation and multiple objects reconstruction; larger capture range; and better calculation complexity. The theoretical contribution is in the development of a new penalty function, used to stop the surface's portions evolvement.

To assure a better 3D visualization we added to the ASS a painting tool, which orders triangles, constructing the object's surface mesh, with respect to their distance to an observer that stays in front of the computer screen [22]. Thus, the closer a triangle to the observer, the lighter the gray level.

The paper is organized as follows: section 2 introduces the continuous 3D active model, while the next section develops the discrete algorithm; section 4 discusses the stability condition that relates the space and time steps in a way to provide a stable convergence to the proper 3D object without self intersection and 3D curls. The truncation error added from derivatives approximation is determined in the same section; a segmentation algorithm is developed in 5; sections 6 and 7 show the obtained results, advantages and bottle necks of the method.

2. THE CONTINUOUS 3D MODEL

Consider the following parabolic differential equation DE [5]:

$$\frac{\partial u(t,p)}{\partial t} = \alpha^2 \frac{d^2 u(t,p)}{dp^2}. \qquad \textbf{(1)}$$

Denote by *C* a closed, smooth, and convex curve parameterized by $C(t,p) = r(t,p) = x(t,p)i + y(t,p)j$ in the domain [-1,1]x[-1,1], where $t \in [0,\infty)$ is a time parameter, which parameterizes the family of curves, whereas $p \in [0,2\pi]$ is a space parameter, which parameterizes the particular curve. Substituting in Eq.(1) the function *u(t,p)* with r(t,p) we receive the geometric heat DE:

$$\frac{\partial r}{\partial t} = \alpha^2 \frac{d^2 r}{dp^2} = \alpha^2 \frac{d}{dp}\frac{dr}{dp} = \alpha^2 \frac{d\vec{T}}{dp} \qquad \textbf{(2)}$$

Taking into account that $\frac{dr}{dp} = \vec{T}$, $\frac{d\vec{T}}{dp} = k\vec{N}$ [24] we arrive at:

$$\frac{\partial r}{\partial t} = \alpha^2 k \vec{N}, \qquad \textbf{(3)}$$

which presents a vector flow [7,8,12,27], defined by the curvature *k* and the normal vector $\vec{N}$. By $\vec{T}$ we denoted the tangent vector to the curve. Extend the vector function *r* from 2D to 3D by the following parametric definition:

$$r(t,\theta,\phi) = x(t,\theta,\phi)i + y(t,\theta,\phi)j + z(t,\theta,\phi)k \qquad \textbf{(4)}$$

The time parameter *t* parameterizes the family of surfaces, whereas θ and ϕ parameterize the particular surface, such that θ parameterizes the surface's curves, which define horizontal planes, and ϕ parameterizes the surface's curves, which define vertical planes. For the 3D function in Eq.(4) we rewrite Eq.(3) in the following form:

$$\frac{\partial r}{\partial t} = \alpha^2 H \vec{N}, \qquad \textbf{(5)}$$

where $\vec{N}$ is the normal vector to the surface defined by Eq.(4), whereas *H* is the mean curvature defined by:

$$H = \frac{k_\theta + k_\phi}{2}, \quad k_\theta = \frac{|r_\theta \times r_{\theta\theta}|}{|r_\theta|^3}, \quad k_\phi = \frac{|r_\phi \times r_{\phi\phi}|}{|r_\phi|^3}. \qquad \textbf{(6)}$$

In Eq.(6) we denoted with k_θ the curvature of the horizontal surface's curves, and with k_ϕ the curvature of the vertical surface's curves, whereas r_θ, and r_ϕ show the partial derivatives with respect to the parameters θ and ϕ respectively. In Eq.(5) the normal vector $\vec{N}$ is calculated by:

$$\vec{N} = \frac{r_\theta \times r_\phi}{\left| r_\theta \times r_\phi \right|} \cdot \tag{7}$$

Denote by P a penalty function to be used by the model to stop evolvement of all surface patches whose normal vectors $\vec{N}$ have reached the surface of a 3D object within the image *I(x,y,z)*. A replacement of α^2 with *P* in Eq.(5) yields:

$$\frac{\partial r}{\partial t} = P\,H\,\vec{N} \cdot \tag{8}$$

It is shown in [24] that the penalty function *P* would take two values 0 and 1. Thus the penalty function could be designed as an implication $P_{\varepsilon,\varepsilon_1}$, which maps the rate of change of the image function *I* (*x, y,z*) on the line interval $\Delta_t \in H\,\vec{N}$ at time *t* to the set $\{0,1\}$ **:**

$$P = P_{\varepsilon,\varepsilon_1}\left(D_{\Delta_t} I = \int_{\Delta_t} \nabla I(x,y,z) \cdot H\,\vec{N}\,dt \right) \to \{0,1\} \tag{9}$$

.

If the rate of change $D_{\Delta_t} I$, calculated on the interval Δ_t, which belongs to the vector $H\,\vec{N}$, is between ε and ε_1, then $P = 0$, otherwise $P = 1$. The numbers ε and ε_1, represent the difference between the background and the foreground of the image and are defined by the user.

3. THE DISCRETE 3D MODEL AND ALGORITHM

To develop the discrete model we use Eq.(8) and employ the following central differences in order to approximate the partial derivatives with respect to *t* and θ [5]:

$$\frac{\partial r}{\partial t} \approx \frac{r(t+\delta,\theta,\phi) - r(t-\delta,\theta,\phi)}{2\delta}$$

$$\frac{\partial r}{\partial \theta} \approx \frac{r(t,\theta+h,\phi) - r(t,\theta-h,\phi)}{2h} \tag{10}$$

$$\frac{\partial^2 r}{\partial \theta^2} \approx \frac{r(t,\theta+h,\phi) - 2r(t,\theta,\phi) + r(t,\theta-h,\phi)}{(2h)^2}$$

Taking into account that the central differences for ϕ are the same as those for θ we receive the following Forward Difference Algorithm (FDA):

$$r(t+\delta,\theta,\phi) \approx r(t-\delta,\theta,\phi) - 2\delta PH(t-\delta,\theta,\phi)\,\vec{N}(t-\delta,\theta,\phi) \tag{11}$$

where H and $\vec{N}$ are computed using Eq. (6) and (7), whose derivatives are approximated by Eqs.(10). Thus, we rewrite Eq.(8) in the following form:

$$r_{ik}^{j+1} = r_{ik}^{j-1} + 2\delta^{j-1} P_{ik}^{j-1} H_{ik}^{j-1}\,\vec{N}_{ik}^{j-1}\,, \tag{12}$$

where *j =1,2,3……* is an index that corresponds to *t* and parameterizes the family of surfaces, whereas *i=1,2,…..,n* and *k=1,2,…..,n* are indexes that correspond to θ and ϕ, and parameterize the horizontal and vertical lines on the particular surface. Denote by *n* the number of normal vectors on each line defined by θ and ϕ, respectively. Follows that, each surface from the family is constructed by mesh containing n^2 points. The number of normal vectors is the same.

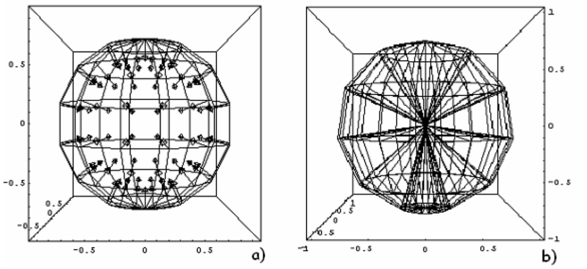

Figure 1. a) The initial surface, which defines a vector field, containing 196 inward normal vectors. b) The normal vectors are extended all the way to the center of the domain.

The discrete form of the penalty function *P,* introduced by Eq.(9), is defined by the following piecewise function:

$$P_{\varepsilon,\varepsilon_1} = P_{ik}^{j}(v) = \begin{cases} 1 \;\; if \;\; PS_{ik}^{j}(v) < \varepsilon \;\; or \;\; PS_{ik}^{j}(v) > \varepsilon_1 \,, \\ 0 \;\; if \;\; otherwise \end{cases} \tag{13}$$

where $\varepsilon < \varepsilon_1$ are thresholds given by the user in order to show the difference between the background and foreground. The penalty sum PS_{ik}^{j} is defined by $PS_{ik}^{j}(v) = \sum_{m=v}^{v+\Delta_t} de_{ik}^{j}(m)$, for $v = 0,...,|\, r_{ik}^{j+1} - r_{ik}^{j-1} \,| - \Delta_t$, where the penalty difference is $de_{ik}^{j}(m) = |\, I_{ik}^{j}(m) - I_{ik}^{j}(m-1) \,|$. In the latter equation $I_{ik}^{j}(m)$ shows the gray level value of the m^{th} voxel that lies on the normal vector at the intersecting point between the i^{th} horizontal and k^{th} vertical surface's lines at time *j*. It follows from Eq. (13), that the algorithm marches $|\, r_{ik}^{j+1} - r_{ik}^{j-1} \,|$ voxels on each normal vector, stops at each voxel *m* and "looks" Δ_t voxels ahead. If the rate of change on the interval Δ_t belongs to [$\varepsilon, \varepsilon_1$], the marching stops and the surface's approximation is set at the m^{th} voxel. Otherwise, go to voxel *m+1* and repeat the algorithm. It follows from the above concepts, that the maximal error of surface approximation, generated by the normal vector force, guided by Eq. (12), is Δ_t.

4. STABILITY CONDITION AND ACCURACY

According to [5] the FDA will stably converge to the solution of Eq.(8) if and only if the time and space steps (δ, h) are related by the following inequality:

$$\delta < \frac{h^2}{2} \cdot \tag{14}$$

To satisfy Inequality (14) we define the space step *h* by:

$$h = \sqrt{\frac{4}{(d_1)^2} + \frac{4}{(d_2)^2} + \frac{4}{(d_3)^2}}\,, \quad \text{and select}$$
$$\delta < 2\sum_{i=1}^{3} \frac{1}{(d_i)^2}\,, \qquad \mathbf{(15)}$$

where d_i, for *i=1,2,3,* denote the sizes of the 3D image. Therefore if we select the values of *h* and δ to satisfy Eqs.(15), we are confident that the initial surface will converge to the proper 3D object's surface without self-intersection and curls.

To determine the accuracy of the FDA we consider the following equations [5]:

$$\frac{\partial r}{\partial t} = \frac{r(t+\delta,\theta,\phi) - r(t-\delta,\theta,\phi)}{2\delta} + O(\delta^2)$$
$$\frac{\partial r}{\partial \theta} = \frac{r(t,\theta + h,\phi) - r(t,\theta - h,\phi)}{2h} + O(h^2) \qquad \mathbf{(16)}$$
$$\frac{\partial^2 r}{\partial \theta^2} = \frac{r(t,\theta+h,\phi) - 2r(t,\theta,\phi) + r(t,\theta-h,\phi)}{(2h)^2} + O(h^4)\cdot$$

Note that we will receive the same equations for the derivatives with respect to ϕ. Using Eqs.(16) we determine that the error added from approximation of the derivatives is on order of $O(\delta^2 + h^4)$, which is better than the error which will be obtained if finite differences are used. The truncation error in the latter case is on order of $O(\delta \ + h^2)$.

Recall that the Forward Difference method was used to develop algorithm (12), but one can employ also backward difference method. In this case, depend on the differences used to approximate the derivatives, he will obtain again $O(\delta^2 + h^4)$ or $O(\delta \ + h^2)$.

5. 3D SHELLS AND SEGMENTATION

As one can tell from the above presentation, the used normal force is a centripetal one. Follows that the ASM is capable of reconstructing the surface of a single 3D object only. Moreover, the object must contain the mass center of the active surface. To overcome this disadvantage and make ASM capable of multiple object reconstruction we developed a 3D scene/image segmentation method as an extension of the one published in [24].

By definition, a 3D image/scene region is called a 3D object, if it consists of one connected set of voxels (points), whose gray level does not exceed a given interval [$\varepsilon, \varepsilon_1$], and whose maximum dimension on a horizontal plane is larger then given threshold Δ_h, maximum dimension on a vertical plane is larger then Δ_v, and the dimension along the normal vectors is greater then Δ_t. The latter we call a height, but Δ_h we call a horizontal radial length, whereas Δ_v is a vertical radial length. In other words, to define a 3D object, Δ_h represents the minimum number of normal vectors which lie in a horizontal plane, and penetrate at least Δ_t voxels into an object. Analogously Δ_v shows the minimum number of normal vectors which lie in a vertical plane and penetrate at least Δ_t voxels into an objec. Now we introduce the notion 3D shell.

Definition1. A 3D shell is said to be a set of voxels

$$Sh\,3D = \{0\bigcup_{u} V_u : V_u = I_{ik}^{0}(u), P_{ik}^{0}(u) = 0 \;\; and$$
$$|\, I_{1j}^{0}(m) - I_{qj}^{0}(m)\,| > \Delta_h, |\, I_{i1}^{0}(m) - I_{ik}^{0}(m)\,| > \Delta_v$$
$$|\, I_{ij}^{0}(m) - I_{ij}^{0}(w)\,| > \Delta_y \;\; for \;\; some \;\; q,k,w \le n\}$$

The Definition1 states that a *3D shell* is constructed by the center of the domain [−1,1]×[−1,1]×[−1,1] and all voxels obtained by the normal vector force and belong to a single 3D object (defined by $\varepsilon, \varepsilon_1$, Δ_h, Δ_v, Δ_t). Thus the shells are composed as conic regions, whose vertex is the origin *O(0,0,0)* and their base is constructed by 3D object's voxels (Fig.2). In the left hand image of Fig.2 is presented a less-permeable groundwater unit, where the 3D shell voxels are denoted by bolded points.

(8

Based on the above theoretical concepts we developed an algorithm to segment a 3D image to shells. The algorithm starts with a convex, smooth and closed surface, which envelops the entire 3D scene/image. Then a normal centripetal force is defined at each intersecting point between horizontal and vertical line. Then the vector force is extended all the way until the center of the 3D scene/image.

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 [25]. 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. Follows that in order to define a shell the minimum number of consecutive horizontal-zero vectors should be Δ_h and the minimum number of consecutive horizontal-zero vectors should be Δ_v.

As we stated above the *shell* algorithm ends if all the vectors in the normal force are zero vectors. Otherwise, the algorithm determines the shells as conic segments defined by set of consecutive horizontal and vertical zero vectors, which satisfy *Definition1* (Fig.2). Thus the normal force splits a 3D scene/image to 3D shells Sh_k, *k=1,2,…..* In each shell the algorithm computes the mass center $\bar{\mu}_k$ and determines the maximal distance *R*$_k$ from $\bar{\mu}_k$ to the shell's voxels. Both *R*$_k$ and $\bar{\mu}_k$ are used as radius and center of initial active surface *C*$_k$ inscribing Sh_k. Further a centripetal normal force, with one and same dimension is defined for each *C*$_k$ and extended all the way to $\bar{\mu}_k$. Then the penalty function is computed for each vector in the force. If the normal force is zero in Sh_k the process of subdivision ends for this shell, which holds a single region, containing the center of Sh_k. If the force is non zero in Sh_k the process of subdivision continues, as described above.

The advantage of *shell* algorithm over conventional level sets and/or explicit algorithms for image segmentation [6,7,8,20] is in the significant reduction of the arithmetic operations, because the *shell* algorithm does not really converge in between the 3D objects.

A disadvantage of the *shell* segmentation is in its inability to handle certain 3D objects distributions in the scene/image. For example, it won't be able to segment an image if multiple 3D objects are presented there and one of them contains the center of the image. Another 3D objects situation that can not be handled is if one of the objects is under the "shadow" of another one toward the direction of the normal force (Fig.3).

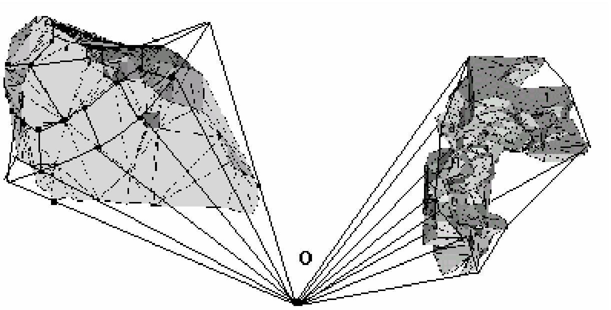

Figure 2. The hand right 3D shell contains 3D less permeable ground water unit. The left hand 3D shell contains a 3D piece of cement.

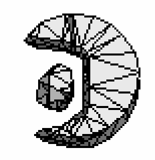

Figure3. A 3D object under the shadow of another one in the toward the direction of the centripetal vector force.

6. IMPLEMENTATION AND RESULTS

The active contour and surface algorithms that can be found in the literature recalculate the normal vectors, and curvatures at each time step [1,6,7,8,12,29]. Although these approaches may produce better approximation of the 3D objects a substantial amount of time is needed to perform the calculations in 3D (remember that the vector flow consists of n^2 members and Eqs.(12) and (13) must be calculated for each vector at every time step). To reduce the number of arithmetic operations our algorithm calculates the inward normal vectors and the curvature at each intersecting point between vertical and horizontal surface lines (at $j=0$), and extends the vectors all the way to the center of the image as is shown in Fig.(1b). We can perform the mentioned simplification, because the stability condition guarantees that the active surface will not change its shape until object's voxels are encountered. Then the algorithm moves a distance $d=|\delta^{j-1}P_{ik}^{j-1}H_{ik}^{j-1}\vec{N}_{ik}^{j-1}|$ along each vector with initial point r_{ij}^{0}. Using Eq.(13) the penalty function searches for voxels that belong to the interval [ε, ε_1], and stops the movement for those vectors, on which at least Δ_t consecutive non-background voxels are encountered. The first one, among the Δ_t foreground voxels, is set to be a surface point. Thus, FDA, introduced by Eq.(12), determines a set M of non-background voxels and uses them to generate a surface mesh, as an approximation of the 3D object's surface. To build up the mesh the algorithm draws a set of horizontal lines linking all voxels $v \in M$ which satisfy the following conditions: the voxels are indexed with one and the same *j, minimum k* for this *j,* and *i=1,2…,n*. To draw the vertical lines the algorithm links all voxels $v \in M$ with one and the same *i*, *minimum j* for this *i*, and *k=1,2,…,n*.

Incorporating the above concepts and using FDA (12), we have developed ASS (active surface software) using *Mathematica 5.0*. The calculation complexity of the ASS is on the order of *O(mn^2j),* where *m* denotes the number of 3D objects in the image, n^2 shows the number of the normal vectors, while *j* denotes the number of the time iterations.

Experiments were performed employing ASS and using input files of the following format $\{x_i, y_i, z_i, gl_i)$, where x_i, y_i, z_i present voxel's coordinates, whereas gl_i shows the gray level value of the voxel. For the first experiment we used data distributed on five closed, arbitrary but parallel curves which construct a surface. As an initial active surface the ASM allows for use any convex surface but we will apply a sphere defined by: $r =< R\sin\phi\cos\theta, R\sin\phi\sin\theta, \cos\phi >$, where $R=\sqrt{2}$. Also $\phi=\frac{2\pi}{n}k$, and $\theta=\frac{2\pi}{n}i$, for *k, i=0,…..,n-1.* The 3D mesh reconstructed by the vector field is given in Fig.4. The upper image provides a view to the top of the surface mesh in the domain [1,1]x[-1,1]x[-1,1], whereas the lower one shows an enlarged front view.

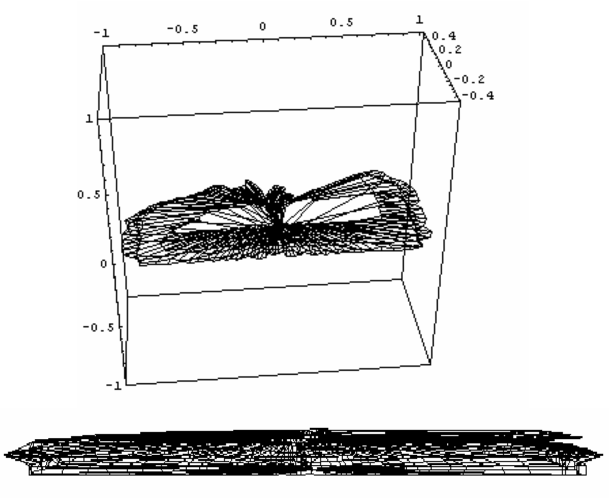

Figure 4. A 3D mesh constructed by 5 arbitrary curves. The upper image shows a view to the top plane, while the lower shows a front view.

For the second experiment we used a file that contains 3D brain data distributed on 6 sections. As an initial active surface we employed a sphere as is shown in Fig 1. The shape of the top section is given in Fig.5(a). The mesh reconstructed by the vector force is shown in Fig.5(b), whereas its enlarged and painted version is given in Fig. 4(c). To perform the experiments we set *n=150*, which yields 22500 vectors in the centripetal force. ASS took less then 20 sec to construct the 3D mesh, and a minute to paint it. A PC with CPU clock frequency 1.8 GNZ was used to perform the experiments.

7. ADVANTAGES AND CONTRIBUTIONS

The ASM and segmentation algorithm, developed in the present paper, has the following advantages over the 3D active surface and level set models given in [1,2,8,10,12,20,29]: larger capture range; faster 3D scene/image segmentation.

The larger capture range (to perform the experiments we used $R=\sqrt{2}$ for the initial sphere, which inscribes the entire domain [1,1]x[-1,1]x[-1,1]) is provided by the ASM introduced by Eq.(8) and the stability convergence condition given by

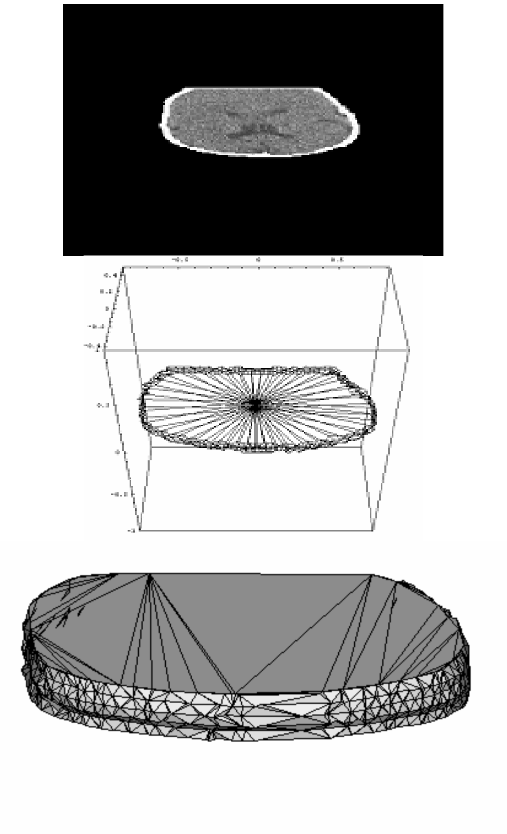

Figure 5. a) A 2D brain section; **b)** The 3D mesh constructed by using 6 consecutive 2D brain sections; **c)** Painted 3D portion of the brain.

Eq.(15). The latter allows for setting the initial surface as far as we want from the 3D objects, unlike some of the methods given in [6,29,31], which must set it close to the targeted 3D object, else the active surface is likely to converge to a wrong shape.

The *shell* algorithm leads to significant reduction of the number of arithmetic operations needed for 3D scene/image segmentation, because it avoids the actual convergences of the active surface between the 3D objects as the conventional active contours and level sets will do [7,10,12,29,31].

Recall that the penalty function is used to halt the evolution for those patches, whose normal vectors encountered a boundary voxels. In [7,12,27] the image gradient together with smoothing Gaussian are used to stop the evolution at the boundary. In practice the discrete gradients are bounded and then the stopping function could be different from zero on a blurred edge. To deal with this problem the isotropic smoothing Gaussian has to be very strong, but it will change the edge too. To overcome these problems [6] proposes an approach based on Mumford-Shah segmentation without stopping edge function (a citation to this technique is given in [6]), but proper coefficients must be determined in this case. To avoid this kind of choosing, to accelerate the evolution process and provide an opportunity for the model to escape a noise with dimensions smaller than Δ_h, Δ_v, Δ_t our method calculates the directional derivative on interval Δ_t in the direction of the normal force.

A good active surface model also based on the heat DE is presented in [8]. The method is capable of 3D reconstruction of single 3D object by using 3D data or 2D data obtained from multiple views. The runtime of the approach is given for a set of experiments but no segmentation features of the method are reported.

A disadvantage of our ASM is its inability to handle spiral objects and objects with complicated concavities [26]. Also, the *shell* algorithm is not able to segment an image if multiple 3D objects are presented there and one of them contains the center of the image. Another difficulty faced by the ASM is to segmented a scene/image where one of the objects is under the "shadow" of another one in the direction of the normal force (Fig.3).

The further work will continue toward development of a solution of the above mentioned problems. A *C++* coding of the entire 3D segmentation, reconstruction and visualization algorithm is to be released.

8. ACKNOWLEDGEMENT

Thanks to REU student Anthony DiPietro for the hard work he did in coding the tool used for 3D mesh definition.

9. REFERENCES

[1] J. Ahlberg, "Active Contours in Three Dimensions", Project #LiTH-ISY-EX-1708, Computer Vision Lab., Linkoping 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, 2000, 486-489.

[3] R. Areste, Y. Yang, J. Hsieh, An Image Enhancement Procedure for 3D Visualization of Liver CT Data, Published by IEEE, Computer Society, 2006, USA, 153-158. *ISBN: 1-4244-0069-4,*

[4] J.Boissonnat, B. Geiger, "Three-dimensional reconstruction of Complex shapes based on the Delaunay triangulation", In Proceedings Biomedical Image Processing and Biomedical Visualisation, In: R. S. Acharya, D.B. Goldgof (Eds.), San Jose CA, International Society of Optical Engineering, Vol.1905, 1993, pp. 964-975.

[5] R. Burden and J. D. Faires, **Numerical Analysis**, 4th Edition}, PWS-KENT Publishing Company, US, 1993.

[6] T. Chan, J. Shen, and L. Vese, "Variational PDE Models in Image Processing", **Notices of the American Math Society**, v.50, n. 1, 2003, pp. 14-26.

[7] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," **International Journal of Computer Vision**, Vol. 22-1, pp. 61-79, 1997

[8] Ye Duan, Liu Yang , Hong Qin, and Dimitris Samaras, "Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces", Proc. of The 8th European Conference on Computer Vision-ECCV 2004, Part III, May 11-14, 2004, Prague, Czech Republic, pp 238 -- 251 (Lecture Notes in Computer Science 3023).

[9] B. Geiger. "Three Dimensional modeling of human organs and its application to diagnosis and surgical planning". **Report de Recherche N^{0}2105**, INRIA, Sophia Antipolis, France, 1994.

[10]L. Gui, X. Bresson, J-P Thiran, Multiscale Image Segmentation Using Active Contours, Ecole Polytechnique Federale de Lausanne, Signal Processing Institute, **Technical Report TR_ITS_2005.012**, Swiss, 2005.

[11] Horace H S Ip, Content-based Retrieval of 3D Models: Feature Extraction and Representation, Proc of the 11th MMM'05, 2005.

[12] S. Kichenassamy, A. Kumar, P. Olver, A., Tannenbaum, and A. Yezzi, "Gradient Vector Flows and Geometric Active Contour Models". **ICCV** 1995: 810-815.

[13] J. Lisani, et al., “Affine Invariant Mathematical Morphology Applied to a Generic Shape Recognition Algorithm”, **Mathematical Morphology and its App. to Image and Signal Processing**, J. Goutsias, L. Vincent, D. S. Bloomberg, eds, Kluwer Pub, 2000, pp. 91-98.

[14] L. Long and G. Thoma, “Segmentation and Feature Extraction of Cervical Spine X-Ray Images”, Proc. of SPIE Medical Imaging: Image Processing, San Diego, CA, February 20-26, v. 3661, 1999, pp. 1037-1046.

[15] H Müller, N Michoux, D Bandon, A Geissbuhler, A review of content-based image retrieval systems in medical applications-clinical benefits and future directions, **Int. J of Medical Informatics**, 2004.

[16] P. Mlsna, N.M. Sirakov, An Intelligent Shape Features Extraction and Indexing System for Fast Medical Image Retrieval, Proc of IEEE Southwest Symposium on Image Analysis and Interpretation, March 28-30, 2004, Lake Tahoe, Nevada, pp. 172-176, 2004.

[17] D. Moody, S. Lozanoff, Surf driver: A practical computer program for generating 3D models of anatomical structures. Proc. of 14th meeting of the American Ass. of Clinical Anatomies, July 8-10,Hawaii, 1997, pp.123-135.

[18] M. Nakazato, T. S. Huang, An Interactive 3D Visualization for Content-Based Image Retrieval, Proc of IEEE International Conference on Multimedia 2001, Tokyo, Japan, August 22-25, 2001.

[19] M. Novotni, B. Klein, 3D Zernike Descriptors for Content Based Shape Retrieval, Proc of *SM’03,* June 16–20, 2003, Seattle, Washington, Published by ACM, USA, 2003.

[20] N. Paragios & R. Deriche. Geodesic, Active Regions and Level Set Methods for Supervised Texture Segmentation. **Inter Journal of Computer Vision**, pp. 223-247, 2002.

[21] J. J. Royer, 3D Modeling and Visualization, 9th Int. CODATA Conference Data Visualization-Earth and Geo Science, Berlin, Germany, 7-10 November 2004.

[22] N.M. Sirakov, F. Muge, A system for reconstructing and visualizing three-dimensional objects, **International Journal Computers & Geosciences**, 27, (1), 59-69, 2001.

[23] N.M. Sirakov, I. Granado, F. Muge, F., “Interpolation Approach for 3D Smooth Reconstruction and Visualization of Subsurface Objects”, **Int. Journal Computers & Geosciences**, Vol.28, Issue 8, pp 877-885, 2002.

[24]Sirakov, N.M.**,** 2006. A New Active Convex Hull Model For Image Regions, **Journal of Mathematical Imaging and Vision**, in printing 05.08.2006, electronic pub 08.14.2006: http://dx.doi.org/10.1007/s10851-006-9004-6

[25]Sirakov, N.M., 2006, Multiple Surfaces Reconstruction from 2D Sections Using an Increasing 2D Vector Flow, In H.R. Arabnia, Xiangjian He, T. Hintz, D. Liu, K.S.Sim (Ed), CSREA Press-USA, June, 2006, Vol. II, pp.364-370. ISBN: 1-932415-93-9, 1-932415-94-7

[26] Sirakov, N.M., Simonelli, I., 2006, A New Automatic Concavities Extraction Model, Published by IEEE, Computer Society, 2006, USA, 178-182. ISBN: 1-4244-0069-4, Library of Congress #: 2005937121

[27] G. Sapiro, “**Geometric Partial Differential Equation and Image Processing**”, Cambridge University Press, 2001.

[28]J.A. Sethian, “A fast marching level set method for monotonically advancing fronts, **Applied Mathematics**, Vol. 93, 02-96, pp.1591-1595.

[29] C. Xu and J. L. Prince, “Snakes, Shaps and Gradient Vector Flow”, **IEEE Tran. on Image Processing**, Vol. 7-3, pp. 359-369, 1998

[30] G. Zamora, H. Sari-Sarraf, S. Mitra, “Estimation of Orientation of Cervical Vertebrae for Segmentation with Active Shape Models”, Proc. of SPIE Medical Imaging: Image Processing, San Diego, CA, February 17-23, 2001.

[31] L. Vese and T. Chan, **“**A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. **International Journal of Computer Vision**, v. 50, n. 3, 2002, pp. 271-293.

[32] Gemcom www.gemcomsoftware.com

[33] GOCAD http://www.ensg.u-nancy.fr/COCAD/

[34] Surf Driver http://www.surfdriver.com