Voxel Grow
A region growing segmentation technique

Ignacio Larrabide, Sebastián Fiorentini
Facultad de Ciencias Exactas, Universidad Nacional del Centro (7000) Tandil
Tel/Fax: (542293) 446552 e-mail: ilarrabi@exa.unicen.edu.ar.

Abstract

The 3D-image segmentation is an important process in the medicine field. Although there are many techniques/algorithms in the literature, there's no appropriate choice that covers all aspects in the field. In the present paper we suggest an alternative region growing method, which performs equally or better than the best-known algorithms. Some tests are shown with real medical data.

Keywords: visualization, computer graphics, segmentation, medical imaging

1. Introduction

The advances in computer technology have transformed in many aspects the way that we treat the information. In the medicine area, the alternatives to capture and analyze data have grown considerably, to achieve the development of tools that make possible the treatment and analysis of 3D images generated by capture devices, like CT (Computer Tomography) or MRI (Magnetic Resonance Image). These types of tools provide assistance to medicine professionals, to facilitate direct analysis over the 3D data in a totally invasive-free manner. Additionally, cheap computers make possible to experiment with solutions that were restricted to high-end computers due to the demand of computational resources and due to the implemented algorithms.

An interesting aspect is the ability to determine certain structures of interest with similar characteristics inside the image using a process known as segmentation. Segmentation constitutes essential stage that conditions the results of the image processing in diverse type of application outposts. For example, in the medicine case, the improvement of different types of practices is feasible, such as planning of surgeries and x-ray treatments, that can be done with greater effectiveness and less risk. In this field there are still many challenges to be surpassed [1]. Many different applications can be found for this kind of image segmentation [6] [7].

Usually, the segmentation demands time and effort for preparation of data by specialized operators. Even, at the present time, it is still usual to do a manual segmentation process through the careful selection of points on the image slices, which constitutes a slow, subjective and non-repeatable work [6].

Segmentation methods can be classified in the following way:

- Pixel intensity based methods. The intensity values of the pixel are used to segment the image. The space continuity is not frequently considered in this type of methods. Within this group, they stand out the method of classification of pixels, which uses a sort of statistical algorithms to assign a label pixel of the image. I.E. [16]

- Region based methods. The image segmentation of is based on the similarity of the adjacent intensities of pixels at image pixels. Within this group they stand out:

    - Region growth methods, where the regions begin being small and soon after, by an iterative process of growth, regions that have similar characteristics are merged [9].

- Model based methods. It begins with a model that grows, updated accordingly to the image characteristics of the image. Within this group, they stand out:

    - Mesh based methods. The model is represented using a mesh, and the update produces changes to it. A typical example of this type of segmentation methods is the algorithms based on snakes. T-Snakes [5] that evolves the desired surface [3] [4] based on mesh approximation concept.
    - Level-Sets based methods: J. T. Sethian initially introduced this method in [8], where Minimum Path principle for curve extraction in the images is used. More in [12] and [13].

Our proposal belongs to region growth strategy and is described in the following section.

2. Voxel Grow

The objective of Voxel Grow method is to look forward directly for the uniform regions on the image, based on local characteristics. Growing is based on a traditional implementation of the traditional growth of region algorithm, taking ideas from [2]. The algorithm can work with one or more seeds, which are input of the method. The concept used is that a same region of the image is homogenous in the intensities of voxels that it contains. Thus, the region that will be taken like interior grows from the seed (s).

2.1. Growth of regions

The growth of the regions is carried out from the seeds that were determined as input, where each one of them contains the following information:

- Position. These are x, y and z coordinates within the image. It is known that this point belongs to the region of interest.

- Intensity. The voxel intensity is important to determine the rank of intensities that will be included in the region (if the inclusion criterion makes use of this value).

Another input data of the algorithm is the three-dimensional image with a cubical matrix shape. The algorithm output will be a matrix with the same dimensions as the input image. This output matrix is initially filled out with zeroes in all the positions, and the seeds will be marked to let the region grow.

Growth Algorithm

An auxiliary FIFO (First In First Out) structure is used where the seeds are initially located, and where the neighbors that belong to the region to be visited are queued up (figure 1).

In algorithm 1 it is possible to see the pseudocode of Voxel Grow algorithm in detail.

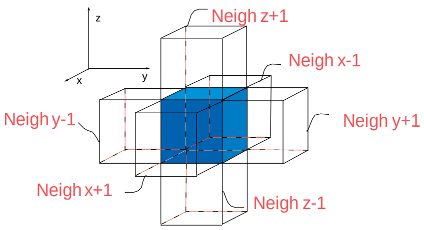


Figure 1. 6-connectedness

The algorithm successively takes elements from the queue. Each one of these elements is one of the volume's voxel that have already been accepted. For each one of them we must visit its neighbors, and decide if that neighbor belongs or not to the region according to the selection criterion [9] (see section 2.2). In order to compare neighbors, 6-connectedness is used. In figure 1 6-connectedness is detailed, where the X, Y and Z neighbors are visited.

Algorithm 1 - Voxel Grow

```
1)  Initialize the auxiliary
volume AuxV to 0
2)  Mark seeds in AuxV
3)  Add seeds to the queue C
4)  While (not C.empty())
  a) Point p =
  C.dequeue_first_elem()
  b) For (each neighbor v of p)
     i)  If (belongs(v))
        (1) mark v in AuxV
        (2) add v to C
     ii) else
       (1)  signal that p doesn't
       belong and was already
       visited
```

One of the most remarkable aspects of this technique is that it always grows by neighbors, so it maintains connectivity between the elements that are included within the segmented region.

The voxel neighbors that have been visited are marked, either belonging or not to the region of interest. Those that belong to the region are marked and queued to visit their neighbors in a following iteration. It is possible to see in the pseudocode, that the algorithm's complexity is O(n).

2.2. Growth Types

Three growth variations are provided to consider if a voxel belongs or not to the region of interest. The first one considers the variation of voxel intensity in relation to the seed intensity. The second one considers the local intensity variation in relation to the neighbor being visited. The last one considers the three-dimensional gradient of the image. These criteria are detailed in the next section.

Seed Growth

In this case the seed intensity is taken always as reference. Each new voxel that is added to the region is included if the intensity difference that exists between it and the intensity of the seed maintains within a threshold determined previously. This threshold is compared directly with the intensity difference. This technique gives as result regions that contain voxels whose intensities are within a certain rank. The function `belongs(Voxel v)` verifies that:

$$|intensity(v)\text{-}intensity(seed)| < \Delta$$

Neighbor Growth

Unlike the previous case, this variation considers that the voxel belongs to the region if the intensity difference with its neighbor remains underneath the threshold. In this technique, voxels that have great variations of intensity with their neighbors are excluded. The next comparison is made:

$$|intensity(v)\text{-}intensity(neighbor)| < \Delta$$

This technique is good to detect well-determined regions, where the limits present a high gradient.

Gradient Growth

From the three techniques that were introduced, this is perhaps the most precise one, because it uses the magnitude of gradient of the image to determine the limits of a region. This value determines how fast the variation of intensity occurs on each voxel of the image. To calculate this magnitude the difference between a voxel and its 6 neighbors is considered (see figure 1). In this kind of growth the threshold determines the gradient magnitude where the region must "stop". The comparison is:

$$|intensity(gradientImage(v_{xyz}))| < \Delta$$

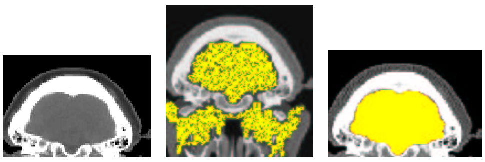
2.3. Region Fusion

The fusion of regions is a major problem in segmentation methods based on growth from a seed. If it is necessary to surround some object, or simply when two fronts of advance come across, the result may have inconsistencies. When two fronts come across, these must be combined. The points from both fronts must be combined and the topology of the region must be modified. In this case, Voxel Grow works on the discreet values of the image, not over a mesh or triangulation of the surface. The problems previously mentioned do not appear since the region grows naturally, surrounding obstacles without having to combine points or calculate intersection of surfaces.

2.4. Draining Control

The mentioned kinds of growth work well in general, but some special cases can fail. For example: when the image has a zone that gradually varies its intensity, arriving too high (or low) values very gradually. In this case the gradient is low between neighbors, allowing the front to advance and "drain" towards zones that do not belong to the region. There are also cases in medical images where small orifices may allow the advancing front to drain towards places that do not belong to the region (see figure 2). Next there are described enhances to the growth algorithm that try to solve this kind of problems.

Instead of growing by means of voxel the growth is made by means of a sphere of voxels. The sphere can have a variable radius, so that a greater sphere is meant to be more restrictive to fit it according to the region that is desired to segment. In some cases it can be desired to segment a zone that is naturally very thin. In these cases it would be appropriate to use a smaller radius, to allow the front to move to small places. Nevertheless for greater and homogeneous regions, using greater radios brings better results. This way the region is segmented completely, avoiding that the advance front propagates toward places that do not correspond to the zone being segmented. The propagation stops when any of the voxels that are being considered does not fulfill the condition of inclusion, or even when the condition of inclusion is fulfilled, the sphere does not enter completely (in other words "it bounces").



(a) (b) (c)

Figure 2. Draining control – (a)Original Image.(b)Without draining control.(c) With draining control

The use of a sphere to cross the volume is very expensive (time consuming). To calculate what voxels are within the sphere of radio r too many operations must be carried out. For that reason a less expensive and simpler alternative was considered: Using a cube variable of size.

Cube adaptation

As it was mentioned previously, the sphere is expensive at run time. Therefore the chosen decision was to develop a growth with a cube shape. This one is based on moving the "walls" of the cube and asking successively for the elements that form each face of the wall. If all the neighboring elements of a wall fulfill the condition of growth, all of those voxels are added to the result. Figure 3 shows voxels included within the cube of radio r = 1. In order to make the growth, the voxels neighboring a complete face are tested. If all of

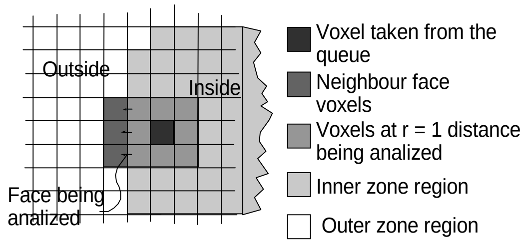


Figure 3. Cube Growth

them are accepted by the inclusion criterion, then all are accepted and queued for a future iteration.

Stretching radius

The control that this technique grants on the growth allows stopping the front of advance to an arbitrary distance of the edge of the region. After the growth, another technique can be applied to advance the last steps. For example, if Voxel Grow was made with radio 3, the limit of the segmented region is 3 voxels away from the limit. At this point a last growth can be applied (with the same criterion of inclusion), using radio 0 (cube face length = 1 + 2*radius) but no more than 3 voxels of distance (it will only get to the limit). Thus the

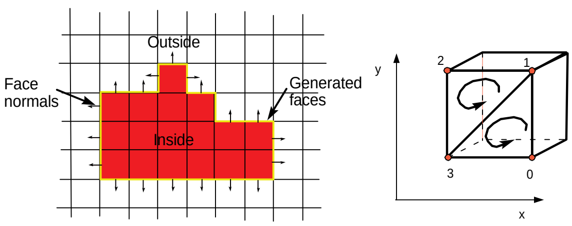


Figure 4. FlatContour face generation

front will not be able to escape to small places, but it will be able to get to the limits of the region.

Advantages

Growing within the regions using basic figures, being either a sphere or a cube, has a series of advantages on the basic growth based on single voxel. The growth under these conditions is more homogenous, since the voxel values being considered are at distances bigger than 1. The changes from one region to another can be gradual and these are not always so well limited. In order to avoid ignoring the limit of a region, which is not very noticeable, and to avoid that the advance front escapes, the voxels taken into account are at greater distance, demanding all the shape voxels to fulfill the growth restriction. These approaches also allow controlling the places by which the advance front can escape. When the propagation is configured with greater radios, the places where the advance front can propagate are more restrictive. This widely improves the algorithm results, since in medical images it is very frequent to notice that small or diffuse zones connect different regions.

Another important aspect is that the algorithm can allow growing from more than one seed. This is important, because there are regions of interest, which there aren't in only one piece (like skull bones). Another interesting point to emphasize is that different inclusion criterion for each other seed can be designated. It is possible to indicate that different seeds grow in a different way according to their location, for arbitrary reasons or because the characteristics of the image vary at different points.

2.5. Surface generation

Usually traditional techniques, like Marching Cubes and Marching Tetrahedra [11], are used for surface generation. These are fast, efficient and work well. Nevertheless, a method called FlatContour, for surface reconstruction was developed. The mesh generation using Flat Contour is a technique for surface reconstruction developed specially to be applied on a segmentation result. The method is based on finding the limits between the segmented zone (or inner) and the outer zone. This algorithm crosses the volume of data (resulting from the segmentation) looking for limits between inner and outer regions (figure 4). Whenever it finds 2 adjacent voxels from different regions, it generates the face of the voxel laying on the limit. Two kinds of limits can be found:

- Inner to outer or
- Outer to inner

The only difference between these two cases is the direction that the generated face must have. The generated triangle faces must always be in direction of the outer zone. As the volume is processed sequentially, the run time is constant, directly depending on the dimensions and not on the amount of triangles. This algorithm generates two triangles for each voxel face. Figure 4 shows the face vertex numeration for the cube and the way linked to create the triangles. The sequence in which the vertices are added can not be arbitrary, because the direction of the triangles depends on these face points, i.e. the direction that will have the normal of the triangle surface.

Surface Post-processing

Once the segmentation and reconstruction by means of Flat Contour is finished, an initial representation of the surface is obtained that limit with the region of interest. In this case the surface follows the contour of voxels included by the segmentation. The mesh triangles are orthogonal with each other. Because of mesh characteristics it is hard to detect the different structures or reconstructed tissues. One alternative to improve the surface quality is to apply a smooth filter to the mesh (see figure 5). This kind of filter works over the mesh points, "relaxing" it and allowing the irregularities to disappear. These irregularities can be interpreted as noise (signals of high frequency). Nevertheless it's very important for the mesh to conserve the topology, because if the mesh is deformed, it will lose details. Considering that the origin of this mesh is a medical image, these details represent parts of the anatomy and they cannot be lost. After making a Fourier analysis over the vertices that compose the mesh, it is

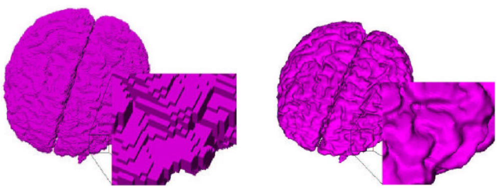


Figure 5. Mesh smoothing result

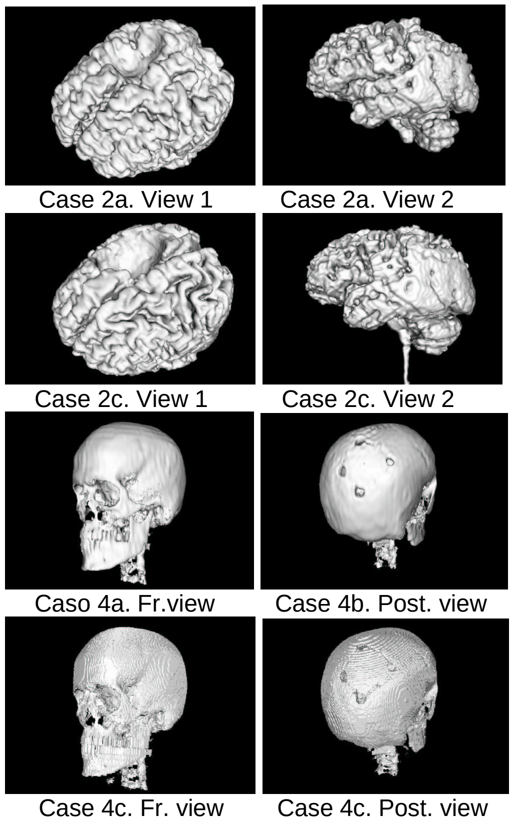
possible to classify them according to the frequency component they have. Then, with this frequency value, it is possible to classify every vertex as a high or low frequency vertex of high or low frequency, and to use this information to determine if it's necessary to move or to leave untouched that particular vertex. The effect is to relax the mesh, improving the form of the cells and distributing the vertices in a more homogenous way. The mathematical algorithm and foundations are described in detail in [10]. Some modifications and optimizations to these methods were also added in [15].

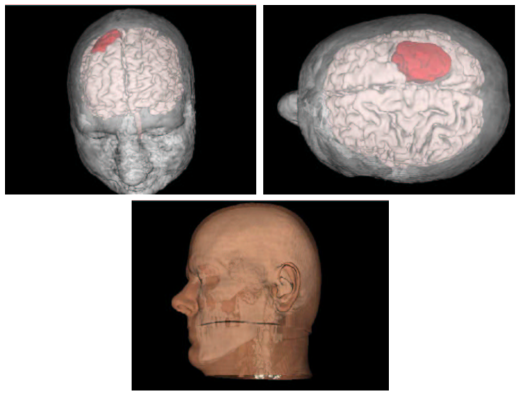
3. Some results

To evaluate the performance a software tool[1] was developed [14] in which the user loads the images, select the seeds, selects a segmentation method and configures it, and a surface is reconstructed. The data considered to make the tests is summarized in table 1.

Type	Case	Dimensions	Tissue
Sphere	1	50 x 50 x 50	-
MRI	2	256 x 256 x 124	Brain
MRI	3	256 x 256 x 124	Tumor
CT	4	256 x 256 x 125	Bone
CT	5	256 x 256 x 125	Skin

Table 1. Test data





Surfaces obtained during tests.

To evaluate the methods, two metrics were used:

[1] The software is available at www.skullydoo.com.ar

1. Time: It is the time needed to segment and reconstruct the surface that represents the segmented data.
2. Surface quality: To evaluate surface quality, number of triangles and NAR (normalized aspect ratio [17]) were taken into account.

The hardware configuration used was an Athlon 900, 512MB Ram, with Windows XP and GeForce2 Video.

Case		Times (ms)		Triangulation quality		
Id	Descrip.[2]	Segm.	Rec.	#Tri	NAR	NAR_a
1a	Mc	961	77	3704	0.577	0.702
1b	Vg+Fc+Tb	961	3275	7020	0.154	0.656
1c	Vg+Fc+C	961	501	7020	0.047	0.570
1d	Vg+Mc	961	190	5768	0.276	0.587
1e	Fm+Mc+T	5999	60	5864	0.169	0.577
2a	Vg+Mc	104671	17153	279088	0.027	0.572
2b	Vg+Fc+Tb	104671	58815	396228	0.001	0.582
2c	Ls+Mc	683833	14776	314362	0.001	0.577
3a	Vg+Mc	104671	13640	16792	0.042	0.570
3b	Vg+Fc+Tb	147191	20321	22928	0.002	0.575
3c	Ls+Mc	843424	10635	12720	0.036	0.570
4a	Vg+Fc+Tb	20880	70852	520216	0.001	0.584
4b	Vg+Mc	20880	11747	428680	0.051	0.569
4c	Mc	0	5107	539050	0.010	0.569
4d	Mc+Tb	0	76029	539050	0.062	0.558
5a	Vg+Fc+Tb	350775	48570	283112	0.003	0.586
5b	Vg+Mc	350775	12609	252588	0.062	0.553
5c	Mc	0	4497	348653	0.007	0.562
5d	Mc+Tb	0	51906	348653	0.001	0.626

Table 2. Tests results

4. Conclusions

In this work, an alternative segmentation method called Voxel Grow was described. Its main advantages are the simplicity, robustness and the quality of its results. These results are comparable to those of more complex and expensive methods like Level Sets with Fast Marching, but improving drastically their run times (in some cases from 6 to 1). Also a method of surface reconstruction called Flat Contour with a smoothing process was developed and it was possible to observe that the combination of Voxel Grow with Flat Contour generate very good quality results.

The method was tested with data from several real medical cases. These data have noise problems and very large amount of information, but the algorithm shows a good performance.

5. Bibliography

[1] J. Duncan, N. Ayache, Medical Image Analysis: Progress over Two Decades and Challenges Ahead, IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol.22, No.1, 2000

[2] D. Fernandez, B. Penesi, F. Ribeiro, M. del Fresno, M. Vénere, Planificación Automatizada de Tratamientos de Radioterapia, Proceedings CACIC´2000, Ushuaia, Argentina, octubre 2000

[3] Medical Image Segmentation Using Topologically Adaptable Surfaces (1997) Tim McInerney, Demetri Terzopoulos - Computer Vision, Virtual Reality and Robotics in Medicine

[4] Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation (1999) - Tim McInerney, Demetri Terzopoulos

[5] Global Minimum for Active Contour Models: A Minimal Path Approach (1997) - Laurent Cohen, Ron Kimmel.

[6] S. Hojjatoleslami, F. Kruggel, Segmentation of Large Brain Lesions, IEEE Transactions on Medical Imaging, Vol.20, No.7, July 2001

[7] Multiscale vessel enhancement filtering (1998) - Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, Max A. Viergever - Lecture Notes in Computer Science

[8] Level Sets methods for Curvature Flow, Image enchancement, and Shape Recovery in Medical Images, June 1995.

[9] K. Castleman, Digital Image Processing, Prentice Hall, 1996

[10] G. Taubin, T. Zhang, and G. Golub. Optimal surface smoothing as filter design. 1996.

[11] Bill Lorensen. Marching cubes: A high resolution 3d surface construction algorithm. In Computer Graphics, Vol.21, No.4, pp. 163-169, 21(4):163–169, 1987.

[12] Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (1998) - J.A. Sethian

[13] Fast Marching Methods (1998) - J.A. Sethian - SIAM Review

[14] Ignacio Larrabide and Sebastián Fiorentini, Digital Image Segmentation with surface models, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, Diciembre 2002.

[15] G. Taubin, Linear Anisotropic Mesh Filters, IBM Research Technical Report RC-22213, October 2001 (under review).

[16] T. K. Moon, "The Expectation-Maximization Algorithm ", IEEE Sig.Proc.Mag., vol. 13, no. 6, pp. 47-60, Nov. 1996.

[17] G.M Treece, R.W. Prager, and A.H. Gee. In Computer Graphics, Vol 21, No.4, pp. 163-149

[2] Vg: Voxel Grow; Ls: Level Sets; Sn: Snakes ; Fc: Flat Contour; Mc: Marching Cubes; Tb: Taubin Smoothing