

# Vertical Coherence Applied to Spect Imagery in the 3D Reconstruction of the Left Ventricle

O Garcia-Panyella<sup>1</sup>, A Susin<sup>2</sup>

<sup>1</sup>La Salle School of Engineering, <sup>2</sup>Technical University of Catalonia, Barcelona, Spain

## Abstract

*As we already presented in the previous edition of the conference, our algorithm uses SPECT data to automatically recover the three-dimensional shape of the human's left ventricle. The reconstruction of both the internal and external surfaces, allowed us to present the synthetic cardiac cycle for an actual patient and derive its associated ejection fraction estimation.*

*The segmentation module has been improved in order to provide a better and robust border detection and classification. Now the filtering process is based on the Maximum-Likelihood Classification approach (MLC), which rewards an edge depending on its probability to be a part of the left ventricle. The algorithm also takes into account vertical coherence between slices that improves the inputs adding information on the 3D shape.*

*In terms of practical results, we test all these new features against several left-ventricle reconstructions applied to actual pathological hearts.*

## 1. Introduction

Bidimensional processing is needed in order to mark the data boundaries of our SPECT imagery. From the 2D data slices, we want to build a 3D voxel dataset that has to be correctly labeled.

From the labeling of the borders, we derive an external force, a vector field [1], which pushes our particle mesh to its final configuration, recovering the 3D desired shape [2].

A first approximation to the 3D modeling would be using a manual process with specific image-processing software like in the segmentation of the visible human dataset [3]. In this huge project, both the human male dataset and partially the female dataset have been segmented by interactive manual procedures by using commercial software.

Automatic or semi-automatic segmentation is an open research task for the biomedical and engineering research community that is giving better results continuously. It gives rise to several difficulties that must be overcome in order to avoid manual processes that require deep medical knowledge and experience.

## 2. The MLC filter

We need to classify the voxels in our initial dataset according to some probability. This probability states if a voxel corresponds to the muscular tissue of the left ventricle.

Given a data slice, there are several generic segmentation methods [4-5], that detect gradient changes, that can be used as the first sight to the edges that we are looking for.

Nevertheless we need some decision rules in order to classify them as belonging to one of the possible classes: external border, internal border or none. See figure 1.

Once the edges have been labeled and classified, we can find the vector field that will act as an external force for the reconstruction phase [1].

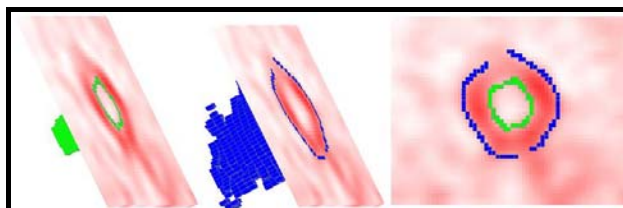


Figure 1. The segmented left ventricle. From left to right: internal (3D), external (3D) and both borders (2D).

Our MLC approach [6] serves as a reliable classification tool. Moreover that, our implementation finds automatically:

- The smallest circle that can be used as a noise removal tool for all the slices. This circle should be more accurate than the manual one defined by physicians.
- The division slice that identifies the beginning of the left ventricle's apex. This slice is calculated first and refined after as explained in section 4.

The algorithm ensures robustness because it performs all the calculations automatically, with no need of symmetry assumptions that might not be achievable due to the input dataset.

Figure 2 presents some results obtained with the MLC filtering algorithm. As it shows, the borders have been correctly detected and labeled.

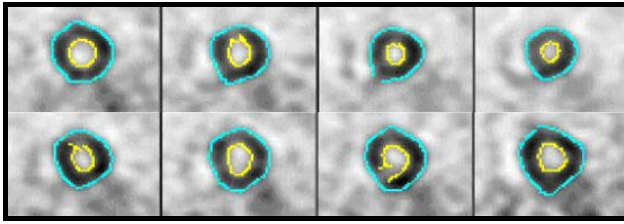


Figure 2. Slice 15 of the same patient during a complete cardiac cycle data set.

The acquisition shown in figure 2 consists on eight captures ranging from systole to diastole. Typical durations for the MLC filtering process are between one and two seconds long (1.484 seconds for every static instant of figure 2, 23.744 seconds in total for all the slices). Each static dataset consists on 94208 voxels (64 x 64 x 23).

### 3. The need for the removal circle

As a prior filter, physicians define a manual circle that acts as a noise-removal tool that deletes everything outside its diameter. It is a key-process for the diagnostic because of its posterior influence on the measures of ejection fraction and wall-thickness. We have automated the circle definition in order to get the smallest one that is still well-fit to data.

However it is important to note that we use the removal circle to cut the contours that the generic edge detector finds, not the property itself.

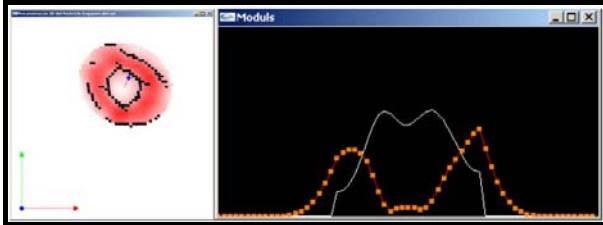


Figure 3. Using the circle for the property cutting off.

If we use the circle filtering over the property values of a slice, we might create false high gradient regions that would be easy to confuse with the real borders. This effect is clearly pointed out in figure 3 where the property for a given section of the data slice is drawn as a white thin line and the gradient is shown by the dotted orange curve. The gradient is maximum at the cut limits although those do not correspond to the data borders, which are closer to the center.

A partial solution might be smoothing the cut data although it is not a desirable method because it blurs all the high gradient changes, including the real borders that we are searching for.

See [6] for more details on the automatic definition of the removal circle.

### 4. Finding and refining the division slice

The division slice downwards determines the passage from two (external and internal) to one (external) surface. It is located at the end of the endocardium (inner surface). Therefore this can be considered an anatomical constraint characteristic from the left ventricle, which is the actual data to be recovered. This constraint is an important feature because it limits the later processes.

In order to find the associated slice, we base on the fact that the property value (blood irrigation) in the global centroid grows up as we go from top to bottom, as depicted in figure 4.

The property is normalized for all the voxels along the dataset. This means that the intensity shown in figure 4 stands for the percentage relative to the maximum. Therefore taking a look at different normalized datasets, gives a definitive clue about typical values for the property differences around the division slice.

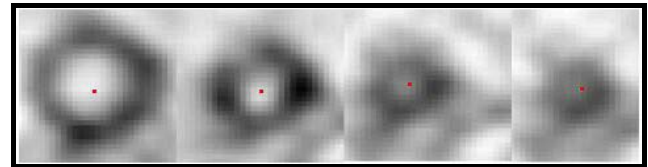


Figure 4. The property at the global centroid grows up (from left to right).

Note that in figure 4, the division slice should be detected between the second and third image.

Our implementation traverses the slices from top to bottom until the division slice is found. The differences in property are tracked until they get expectedly big. This high property slice determines the endocardium bottom and therefore the division slice.

The division slice is refined to ensure its correctness. The refinement consists on taking advantage of the vertical coherence that should characterize our dataset.

If we are consistent with our knowledge of the left ventricle, we must know that there are several cases that are very unlikely to happen. Those cases can be derived from the labeled borders in the slices.

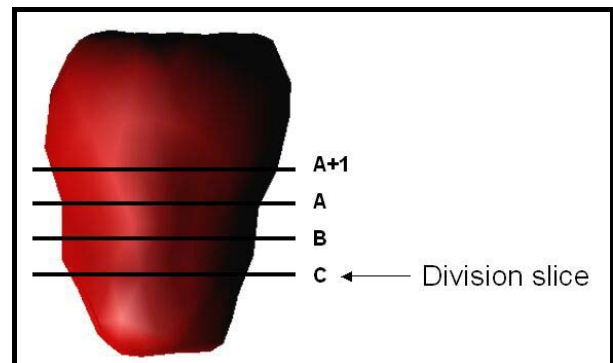


Figure 5. The division slice refinement process.

In figure 5, C is the division slice that we found before the refinement. A + 1, A and B are the slices in the top of it. The technique examines the labelings in the first three slices (A, B and C) to see if everything is coherent. It also examines the labeled borders of slice A + 1 when necessary.

By examining the labelings we mean checking if the slice contains one (external) or two (external and internal) borders.

Table 1. Cases to take into consideration when refining the division slice.

Slice/Case	1	2	3	4	5	6	7	8
A	1	1	1	1	2	2	2	2
B	1	1	2	2	1	1	2	2
C	1	2	1	2	1	2	1	2

Table 1 shows the eight cases that arise. For every case, we can see how many borders are found in slices A, B and C. If there is only one border, we understand that it has to be external because we are in the apex area; if not we should have two borders, where one is internal and the other is external. Let us examine each case separately:

1. Very unlikely to happen because it means that the apex is formed from lots of slices. This fact is totally uncommon. Notify it to the user.
2. Not possible if the quality of the data is within some certain limits. We have two borders when beginning the apex and only one in the top of it. Notify it to the user.
3. Neither A or B where incorrectly catalogued. We must check A + 1:
  - a. If A + 1 contains two borders, the division slice is still considered to be C because we consider that A was incorrectly catalogued with a unique border when it should have two.
  - b. If not, we do not have a possible situation if the quality of the data is within some certain limits. Notify it to the user.
4. Not possible if the quality of the data is within some certain limits. Notify it to the user.
5. B is the final division slice.
6. Not possible if the quality of the data is within some certain limits. Notify it to the user.
7. C is the final division slice so no changes have to be made.
8. Not possible if the quality of the data is within some certain limits. Notify it to the user.

In figure 6, the case has been labeled as 7. The algorithm outputs 2, 2 and 1 for the slices A, B and C respectively. As the images show, it is clear that slice C contains one unique external border while A and B contain two borders, one external and the other internal.

Then the division slice was correctly labeled at the first process and no corrections need to be made.

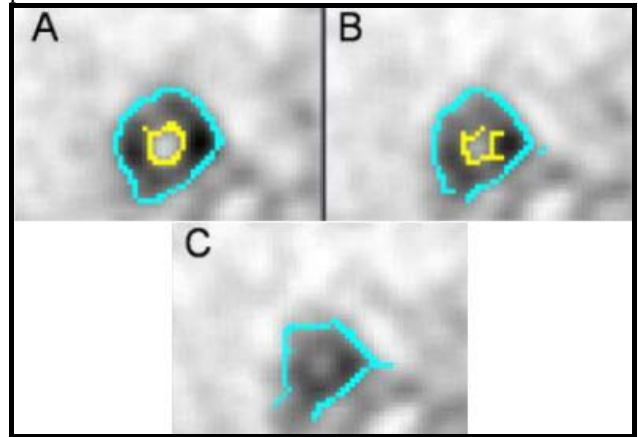


Figure 6. Case 7 of the division slice refinement process.

## 5. Bounding Box based filtering

After the previous filters, there are several contours that might “survive” and that are not associated to the real data borders. Besides that, those contours might be inside the automatic circle and big enough to avoid the consideration of spurious.

The Bounding Box based filter is designed for taking ride of those. This filter relies totally in one of the slices, the slice that contained the biggest external contour that was used for finding the automatic circle [6]. It assumes that this is the biggest slice in terms of useful data, which should be true if the previous processes where successful. For the seek of clearness, we will name this slice, the big slice.

The filter finds the circular bounding boxes associated to the borders of the big slice. Those bounding boxes are used to filter the upper and lower slices as it follows:

- From the big slice to the division slice: we traverse all the slices by using the actual’s slice bounding boxes (internal and external) as a noise-removal tool over all the subsequent slices.
- From the big slice to the top: all the slices are filtered with the bounding boxes (internal and external) of the big slice.
- From the division slice to the bottom: we traverse all the slices by using the current slice bounding box (only external) as a noise-removal tool over all the subsequent slices.

Figure 7 shows an actual case where the big slice and its circular bounding boxes are shown. Both are rendered in cyan (external) and yellow (internal).

The external border (strong blue) of this slice is the biggest within this dataset.

All the calculations were performed automatically.

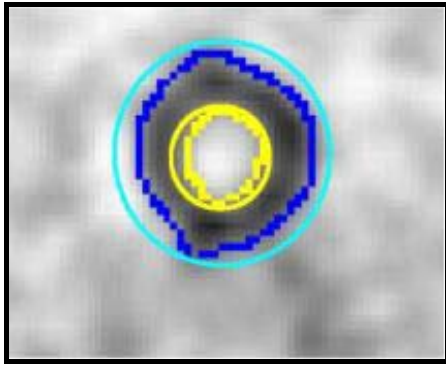


Figure 7. Using the bounding boxes filter in real data.

## 6. Results

We present six pathological cases that have been treated with the proposed algorithms. All of them were simulated using the Runge-Kutta 4 solver, the free deformation model and a final smoothing process [6]. The tests were performed in a Pentium III PC with 256 MB RAM and a 32 MB accelerator card (mobility ATI Radeon).

Figure 8 shows two datasets of ventricles originally affected by digestive activity in their apex. However, the algorithm can distinguish correctly the left ventricle borders. Note how the surfaces are well-fit to the voxelized data to recover. The simulations took 0.56 (top) and 0.45 (bottom) seconds.

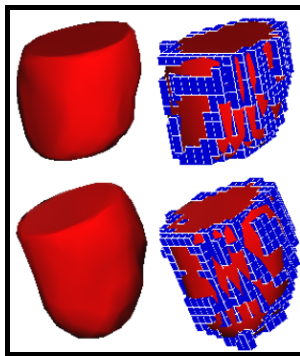


Figure 8. Digestive activity perturbing initial datasets. Left, final external surfaces; Right, final surfaces over the dataset to recover.

Figure 9 shows four more cases. Their respective initial datasets presented occlusion defects located at several places.

The defects were classified by physicians as inferior (1), apical (2), side (3) and vast anteroapical (4). The simulation times took 0.32, 0.45, 0.44 and 0.68 seconds respectively.

Note how the lack of information in the original datasets influences the final reconstructed surfaces (cases 3 and 4).

The final smoothing process avoids the aparition of

creases along the mesh.

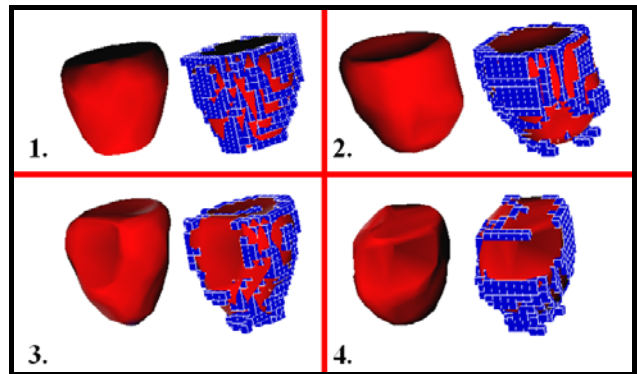


Figure 9. Oclusions perturbing initial datasets. For each case the final external surface is rendered in the left and mixed with the dataset to recover in the right.

## Acknowledgements

This work has been partially financed by the TIC2000-1009 project. The first author is granted by an EPSON “Rosina Ribalta” prize.

The authors thank the medical applications research group at the IRI-UPC, the Nuclear Cardiology team at Vall d’Hebrón Hospitals and the people integrating the graphics & VR group at the Multimedia Section of La Salle School of Engineering.

## References

- [1] Xu C, Prince JL. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing* 1998;7(3):359-369.
- [2] Garcia O, Susin A. Left Ventricle Volume Estimation from 3D SPECT Reconstruction. *IEEE Computers in Cardiology* 2002;29:621-624.
- [3] Quackenbush D, Ratiu P, Kerr J. Segmentation of the Visible Human Data Sets. Available at: URL: [http://www.nlm.nih.gov/research/visible/vhp\\_conf/quackenb/quackenb.htm](http://www.nlm.nih.gov/research/visible/vhp_conf/quackenb/quackenb.htm). Accessed July 20, 2003.
- [4] Canny J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1986; 8(6):679-698.
- [5] Ruzon M, Tomasi C. Color Edge Detection with the Compass Operator. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1999;2:160-166.
- [6] Garcia O, Susin A. MLC filtering applied to the 3D reconstruction of the left ventricle. *CEIG 2003, XIII Congreso Español de Informática Gráfica* 2003;13:17-30.

Address for correspondence.

<sup>1</sup>La Salle School of Engineering.  
C/ Quatre Camins,2. 08022 Barcelona, Spain, Europe.  
oscarg@salleURL.edu - <http://www.salleurl.edu/~oscarg/>

<sup>2</sup>Departament Matemàtica Aplicada.  
Av/ Diagonal, 647. 08028 Barcelona, Spain, Europe.  
toni.susin@upc.es - <http://www-ma1.upc.es/~susin/>