An effecient automatic segmentation of left ventricle by hybrid approach

M. Arsalan KHAWAJA* and Bui Thien Bao†
*Center Universitaire Condorcet,
Universite de Bourgogne,
71200, Le Creusot,
France.
Email : Muhammad-Arsalan_Khawaja@etu.u-bourgogne.fr
†Center Universitaire Condorcet,
Universite de Bourgogne,
71200, Le Creusot,
France.
Email :

***Abstract*—Left Ventricle is an integral part of the heart. It pumps blood into the body. Cardiovascular diseaes are quite common in modern world. The health of the heart can be studied using left ventricle. Most of the heart diseases can be diagnosed by retrieving important information from systole and diastole operations of heart via effective medical imaging analysis of Left Ventricle (LV). The segmentation of Left Ventricle plays an important role in retrieving the ejection fraction (a parameter which indicates the heart's health). Since Magnetic Resonance Images (MRI) are noisy and not really perfect, scientific community is always curious to find more effecient, reliable, robust and more accurate segmentation methods to calculate ejection fraction. In this research project, we use a novel hybrid approach for segmentation. K-means, along with border following algorithm is used for segmentation. The results are quite impressive and discussed in the last section.**

***Index Terms*—K-means, Python, Border Following Algorithm, Left Ventricle, Adaptive Smoothing, Clustering, Hough Transform.**

I. Introduction

The segmentation of left ventricle is of utmost importance when it come to diagnosis of heart diseases. It takes too much of a time for manual segmentation and is a painstaking task. This segmentation establishes and provides us with important knowledge of critical parameters which can help us calculate certain ratios. These ratios help doctors to predict the condition of heart and in some cases diagnose a particular disease. In short, an efficeint segmentation of heart ends up saving patient life.

Unfortunately doctors are too busy to manually segment each of the medical image and it also wastes alot of thier time. They are more needed to patients. There is an inevitable demand for some automatic segmentation of left ventricle. Alot of work has been done in this field, but because medical images are not really the most perfect images, segmentation algorithms are destined to struggle with accuracy. Some of the most common methods used for demonstrated in figure 1.

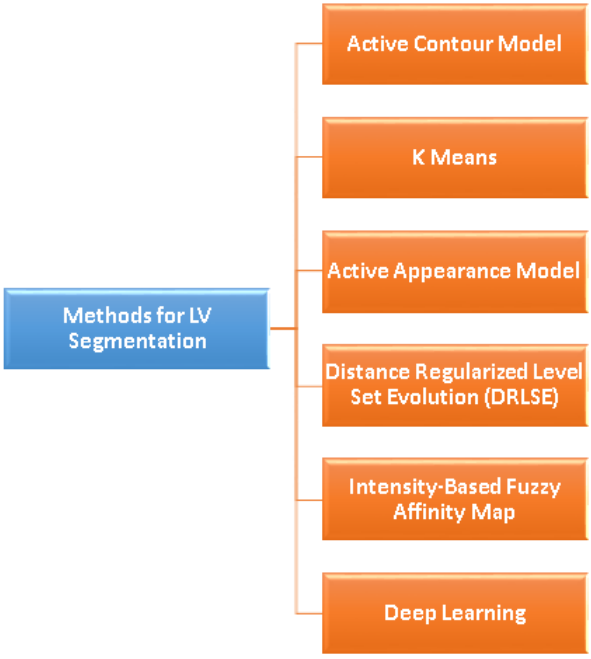

Figure 1. These are some methods / Algorithms used to segment left ventricle. Each of them have their own advantages and disadvatages. None of them is perfect.

We went through the algorithms and decided to use Hybrid Appoach for our segmentation. In hybrid approach, we intend to use more than one approaches / algorithms to work simaltaneosly on the segmentation of the image. In this project we used mainly primarily K-Means algorithm inspired by [5]. Alongside, a border following algorithm inspired from [7] was used to enhance segmentation results. The preproccessing was done with adaptive smoothing algorithm. Finally we optimized the performance of our algorithm by repeated experiments of using the algorithm with different datasets and setting the parameters to optimal value.

II. Anatomy of Heart

The heart is made up of four chambers: two upper chambers known as the left atrium and right atrium and two lower chambers called the left and right ventricles. These are shown in following figure .

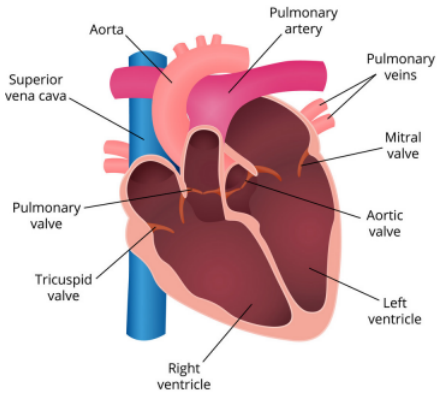

Figure 2. The left ventricle is longer and more conical in shape. In short axis, the left ventricle appears circular in shape. The image is taken from [1]

The left ventricle is considered an important part of the cardiovascular system. It is thought off as a pump that supplies blood to the body. The mass of the left ventricle, as estimated by magnetic resonance imaging, averages $143g \pm 38.4g$, with a range of 87–$224g$ [3].

Another important thing to discuss are the two phases of the cardiac cycle. They are called Systole and Diastole. They occur as the heart beats, pumping blood through a system of blood vessels that carry blood to every part of the body. Systole occurs when the heart contracts to pump blood out, and diastole occurs when the heart relaxes after contraction.

A. Ejection Fraction

Ejection fraction (EF) refers to how well your left ventricle (or right ventricle) pumps blood with each heart beat [5]. The ejection fraction can be calculated as follows:

$$\mathrm{EF} = \frac{V_{\mathrm{endo}}\left(t_{\mathrm{D}}\right) - V_{\mathrm{endo}}(\mathrm{ts})}{V_{\mathrm{endo}}\left(t_{\mathrm{D}}\right)} \tag{1}$$

Here note that V_{endo} is the volume of the inner walls of the heart, $V_{endo}(t_D) = max_t[V_{endo}(t)]$ is the end-diastolic volume and$V_{endo}(t_S) = min_t[V_{endo}(t)]$ is the end-systolic volume.

III. Methodology

The first task was to identify the tasks in the project. After we were done with identifying the tasks, we came to their implementation on python. We studied and learned to use the needed libraries in python. The libraries hae been given in Appendix A. The following flowchart gives a very articulate work flow 3 demonstration.

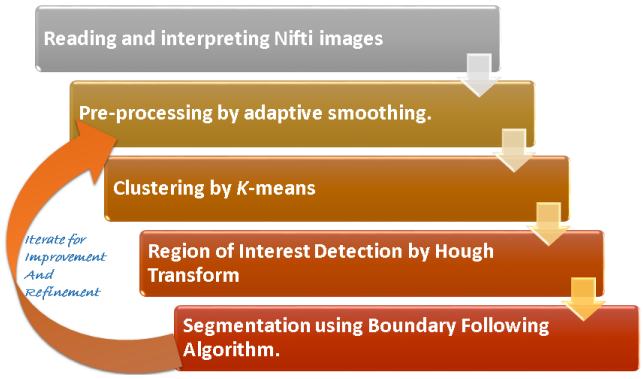

Figure 3. The flowchart demonstrates work flow. Neccessary iterations were done to find the optimal combination of all algorithm parameters with the goal of acheiving the highest accuracy possible.

IV. Development

In this section we will discuss the implementation of algorithm. Python programming language was used for the implementation. The methodology from figure 3 was observed to complete the project. We faced challenges in almost every stage but our resiliance and motivation kept us going. The following subsections take us step by step through the ladder of project.

A. Adaptive non linear smoothing

We considered Feature-Preserving Adaptive Smoothing Method (FPASM) for our project. The primary K-Means research [5] that we were following took this algorithm from another research [2]. Adaptive smoothing is a class of typical nonlinear smoothing techniques that has been studied for many years.

The idea is to adapt pixel intensities to the local attributes of an image on the basis of discontinuity measures. Note that in this algorithm there are two kind of discontinuities, Local and Contextual discontinuity. This novel approach joins these both discontinuities to preserve edges and at same time smooths images. It is brilliant method according to us.

In order to measure local discontinuities, we define four detectors for pixel (x, y) along four directions. These four directions are vertical (V), horizontal (H), diagonal (D), and counter-diagonal (C), respectively. Accordingly, four detectors are defined as [2]

$$\begin{aligned} E_{H_{xy}} &= |I_{x+1,y} - I_{x-1,y}| \\ E_{V_{xy}} &= |I_{x,y+1} - I_{x,y-1}| \\ E_{C_{xy}} &= |I_{x+1,y+1} - I_{x-1,y-1}| \\ E_{D_{xy}} &= |I_{x+1,y-1} - I_{x-1,y+1}| \end{aligned} \tag{2}$$

Here $I_{(x,y)}$ is the intensity of pixel (x, y). To be more clear we constructed this figure 4.

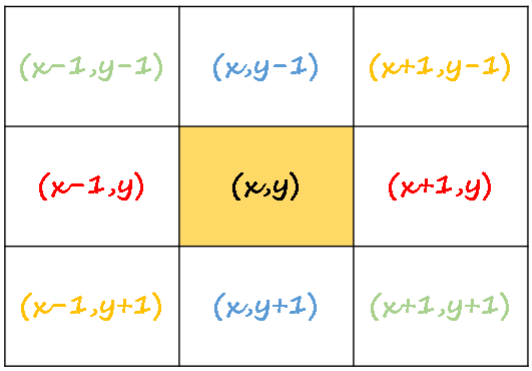

Figure 4. The pixel (x, y) is our pixel of operation. We see the local discontinuity defined in equations 32 is nothing but calculating the differences around the neighbouring pixels.

With these diferences in mind, we define the local discontinuity as [2]:

$$E_{xy} = \frac{E_{H_{xy}} + E_{V_{xy}} + E_{C_{xy}} + E_{D_{xy}}}{4} \tag{3}$$

In order to detect contextual discontinuities, we use spatial variance to make a measure. First, we define a contextual neighborhood associated with pixel $(x; y)$, $Nxy(R)$,as [2]:

$$N_{xy}(R) = \{(i, j)|x-R \leq i \leq x+R, y-R \leq j \leq y+R\} \tag{4}$$

Note that, here $R(R > 1)$ is a parameter that determines the size of this contextual neighborhood. The contextual neighborhood is defined here without explicitly counting image boundaries. The parameter R describes a spatial scale or a resolution that critically determines results of the contextual discontinuity measure. We calculate the mean of pixels on $N_{xy}(R)$, $\mu_{xy}(R)$, as [2]:

$$\mu_{xy}(R) = \frac{\sum_{(i,j)\in N_{xy}(R)} I_{i,j}}{|N_{xy}(R)|} \tag{5}$$

and the spatial variance, $\sigma^2_{xy}(R)$, can be calculated as [2]:

$$\begin{aligned}\sigma^2_{xy}(R) &= \frac{\sum_{(i,j)\in N_{xy}(R)} \left(I_{i,j} - \mu_{ij}(R)\right)^2}{|N_{xy}(R)|} \\ &= \frac{\sum_{(i,j)\in N_{xy}(R)} I^2_{i,j}}{|N_{xy}(R)|} - \left(\frac{\sum_{(i,j)\in N_{xy}(R)} I_{i,j}}{|N_{xy}(R)|}\right)^2\end{aligned} \tag{6}$$

The normalized $\tilde{\sigma}^2_{xy}(R)$ can be written as [2]:

$$\tilde{\sigma}^2_{xy}(R) = \frac{\sigma^2_{xy}(R) - \sigma^2_{\min}(R)}{\sigma^2_{\max}(R) - \sigma^2_{\min}(R)} \tag{7}$$

where$\sigma^2_{\max}(R)$and $\sigma^2_{\min}(R)$are the maximal and minimal spatial variance across the entire image, respectively. Intuitively, $\tilde{\sigma}^2_{xy}(R)$ reflects the relative degree of the contextual discontinuities for pixel $(x; y)$. The table I describes the influence of $\tilde{\sigma}^2_{xy}(R)$ in a more articulate way on the smoothing.

Parameter	Value set	**Indication**
$\tilde{\sigma}^2_{xy}(R)$	Large Value	presence of significant feature
	Small Value	No significant feature or significant edges found
	$< \theta_\sigma$	usually it means noise
	$>\theta_\sigma$	it's a feature
α	Small value	fast smoothing and discontinuity reduction
	Large value	slow smoothing so that important features can be preserved
S	Small value	leads to better preservation of details
	Large value	all discontinuities disappear

Table I
ANALYSIS OF CHANGING SMOOTHING PARAMETERS

So, these results reccomend us that the local attributes of a pixel with a high contextual discontinuity should be preserved and those of a pixel with a low contextual discontinuity should be smoothed toward homogeneity. However, both noise and trivial features irrelevant to a given problem lead to the complexity for visual information processing. In order to reduce their influence in the contextual discontinuity estimation, we introduce a transformation into $\tilde{\sigma}^2_{xy}(R)$[2];

$$\Phi\left(\tilde{\sigma}^2_{xy}(R), \theta_\sigma\right) = \begin{cases} 0 & \tilde{\sigma}^2_{xy}(R) < \theta_\sigma \\ \tilde{\sigma}^2_{xy}(R) & \tilde{\sigma}^2_{xy}(R) \geq \theta_\sigma \end{cases} \tag{8}$$

Here $\theta_\sigma\,(0 \leq \theta_\sigma \leq 1)$ is a threshold. For a given $\theta_\sigma, \Phi\left(\tilde{\sigma}^2_{\pi v}(R), \theta_\sigma\right)$ gives us lead to contextual discontinuity map.In summary, pixels of same area or region are supposed to have low contextual discontinuities except the case, for those pixels located near boundaries that have high contextual discontinuities. Unlike local discontinuity measure, conextual discontinuity measure is relatively insensitive to local intensity changes.

Now that we have established the concepts of Local Discontinuity and Contextual discontinuity, It is easier to understand the smoothing algorithm. The key concept of adaptive smoothing is to update a pixel's intensity through the influence of local weighted averaging of its neighboring pixels' intensities. The flow chart in figure 5 gives the birds eye view of smoothing algorithm.

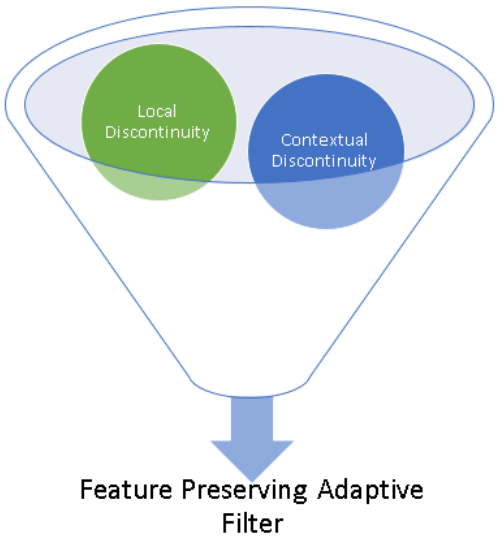

Figure 5. Flow Chart for the idea of Adaptive filter. In step-1, we calculate local discontnuity. In step-2, we calculate contextual discontinuity. In step-3 with the suitable guess of parameters R and θ , we use equation 9 to calculate filtered image. If the results are not satisfactory, we play with the parameters and reclculate the filtered image upto reasonable nunmber of iterations.

The smoothing algorithm as shown in equation 9 runs through the entire image updating each pixels intensity value I^t_{xy}, where t is the iteration value [2].

$$I^{t+1}_{xy} = I^t_{xy} + \eta_{xy} \frac{\Sigma_{(i,j)\in N_{xy}(1)/[(x,y)]} \eta_{ij} y^t_{ij} \left(I^t_{i,j} - I^t_{x,y}\right)}{\Sigma_{(i,j)\in N_{xy}(1)/[(x,y)\}} \eta_{ij} \gamma^t_{ij}} \quad (9)$$

here note that α is a parameter whose value we set according to table I:

$$\begin{aligned} \eta_{ij} &= \exp\left(-\alpha \not\nabla \left(\tilde{\sigma}^2_{xy}(R), \theta_\sigma\right)\right) \\ \gamma^t_{ij} &= \exp\left(-E^t_{ij}/S\right) \end{aligned} \quad (10)$$

The main goal of smoothing is to alleviate the complexity for subsequent processes in early vision. With this pre- processing, the results of our MRI image are shown in figure 6.

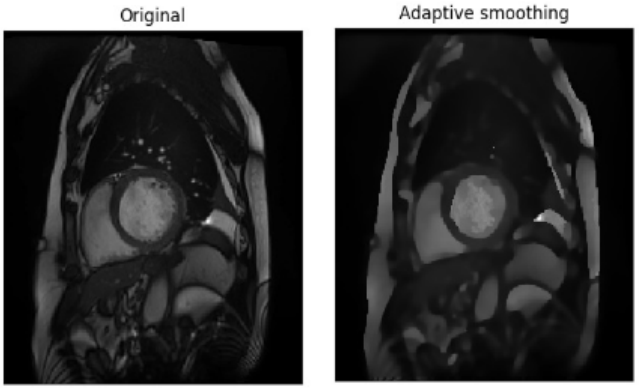

Figure 6. On the left, is the original image. On the right is the smoothed image. We can clearly notice that the algorithm has blurred the details of object but takes care of the edges. This will help us in getting great accuracy for segmentation.

B. K-Means Clustering

The smoothed images were then clustered using k-means algorithm proposed by Duda and Hart [4], [6]. This algorithm has four steps as shown in figure 7 to search for the image clusters.

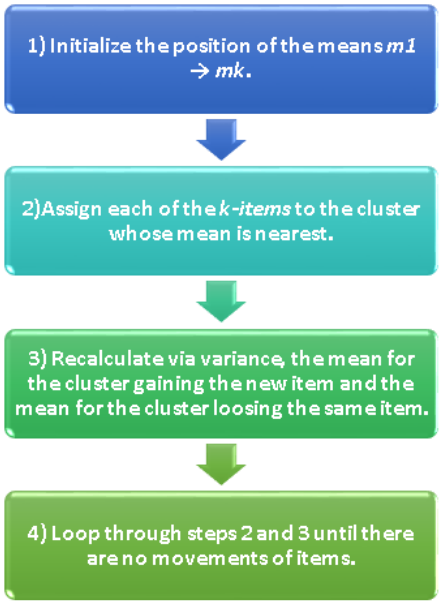

Figure 7. The figure demonstrates the process of K-Means algorithm step by step. The algorithm is quite simple and is widely used in Machine learning, Econmics, Data science and many other fields.

The key idea of the K -means algorithm is to divide M points in N dimensions into K clusters, such that the within-cluster sum of squares is minimized [4] .K-means Clustering is categorized as unsupervised learning, which means that it does not need any intervention by human in our project and leads way to automatic segmentation of Left Ventricle. The figure 8 shows results after applying k-means clustering.

Figure 8. Clustering of the original image as shown in figure 6. The image has been clustered using K means algorithm with $k = 2$, where k is the number of clusters.

C. Detection of Region of Interest (ROI)

It was a trickiest part of the project. We tried a number of approaches to detect the region of interest, i.e the region

containing left ventricle and its nearby surroundings. The approaches have been defined described briefly:

1) Hough Transform: We know that the left ventricle can thought of as an approximation of circle. Hough transform is used to detect circular features in an image. We applied hough transform and detected ventricle. With the centre of ventricle detected as centre of circle, we declared the reasonable area around centre of circle as region of interest. However this approach was not a great idea because:

- Left ventricle is the only circular shape object in image, Many slices had more than one circular object other than left ventricle which allowed it hough transform to detect false left ventricle.
- In some images / slices, left ventricle becomes too dark to be detected as an object.
- In some images / slices, the left ventricle is not near to circular shape. so hough transform fails

D. Segmentation of Left Ventricle

A border following algorithm inspired by [7] was used to segment Left Ventricle. Border following is one of the most important and popular technique in segmentation of binary images. . It derives a sequence of the coordinates or the chain codes from the border between a connected component of l-pixels[1] (l-component) and a connected component of O-pixels[2] (background or hole).

The basic idea of border following algorithm is simple. It works on binary image and we already produced binary image by $k = 2$ (2 clusters) with k-means clustering. The border following algorithm as illustrated in figure 9 sees th component as 1 and background as 0. When it moves through the pixel grid passing each pixel watching out their intensity (0 or 1). As soon as it observes a jump from O-Pixel to 1-Pixel, it memorizes the pixel location and value. After it memorizes[3] all the jumps i.e switching from O-Pixel to 1-Pixel and vice versa, It categorizes the the border in number of objects depending whether the pixel locations are connected to each other or they are sparsely located.

[1]Pixels with densities 1 are called the 1-pixel
[2]Pixels with densities 0 are called the O-pixel
[3]By memorize we mean it saves the pixel location in a vector

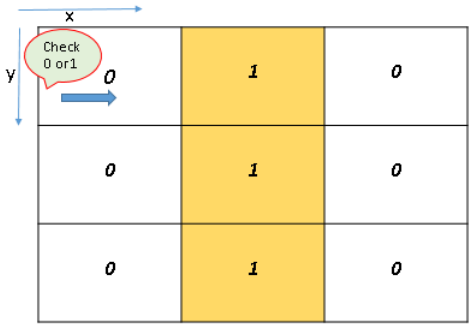

Figure 9. The simple 3×3 image demonstrates the border following algorithm. The yellow pixels are some object border. The algorithm moves through the pixel grid looking for the jump in pixel intensities. It memorizes that jump and categorizes them to respective objects according to their spacial location.

The resulting segmented image is shown in figure10.

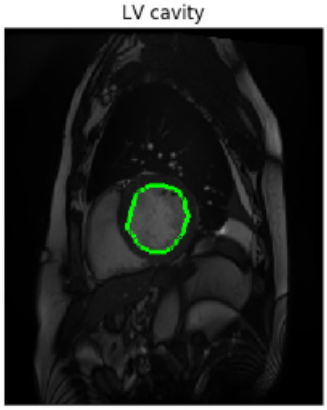

Figure 10. Final segmented image. The left ventricle is indicated by green contour. The original image is shown in figure 6

V. Improvements and Refinements

initial accuracy , changes made, accuracy increased,

A. First Version

discuss initial version

B. Second Version

discuss final version

VI. Final Results

discuss analyze

VII. Executive Summary

In this project we used mainly primarily K-Means algorithm inspired by [5]. Alongside, a border following algorithm inspired from [7] was used to enhance segmentation results. The preproccessing was done with adaptive smoothing algorithm. Finally we optimized the performance of our algorithm by repeated experiments of using the algorithm with different datasets and setting the parameters to optimal value.

VIII. User Guide

how to use app.

References

[1] Vector stock: Anatomy of the human heart vector image.

[2] Ke Chen. A feature preserving adaptive smoothing method for early vision. *National Laboratory of Machine Perception, Peking University, China, Technical report*, 1999.

[3] Jessica S Coogan, Jay D Humphrey, and C Alberto Figueroa. Computational simulations of hemodynamic changes within thoracic, coronary, and cerebral arteries following early wall remodeling in response to distal aortic coarctation. *Biomechanics and modeling in mechanobiology*, 12(1):79–93, 2013.

[4] Hartigan. A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[5] Michael Lynch, Ovidiu Ghita, and Paul F Whelan. Automatic segmentation of the left ventricle cavity and myocardium in mri data. *Computers in biology and medicine*, 36(4):389–407, 2006.

[6] Duda Richard. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[7] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

Appendix A : Python Libraries Used