

# Automatic Segmentation of Cardiac MRI Using Snakes and Genetic Algorithms

Gustavo Miranda Teixeira, Igor Ramalho Pommeranzembaum,  
Bernardo Lino de Oliveira, Marcelo Lobosco, and Rodrigo Weber dos Santos

Departamento de Ciência da Computação, Universidade Federal de Juiz de Fora (UFJF),  
Juiz de Fora, Minas Gerais, Brasil  
magusbr@gmail.com, tosobaum@yahoo.com.br, belino@acessa.com,  
{marcelo.lobosco,rodrigo.weber}@ufjf.edu.br

**Abstract.** In this work we study and implement techniques for the automatic segmentation of cardiac Magnetic Resonance Images. The methods are based on the active contours algorithm called Snakes, which are adapted and tailored to the specific task of automatic segmentation of the left ventricle of the heart in Magnetic Resonance Images. We propose a new external force to improve the convergence of the Snakes method. In addition, a genetic algorithm is used to find the best set of configuration parameters for the Snakes method. The algorithms are implemented in Java and threads are used to explore data parallelism on shared-memory machines. Tests are performed on 150 short-axis images acquired from two healthy volunteers. Preliminary results suggest the proposed methods are promising and with further development and validation may be used, for instance, for the automatic calculation of cardiac ejection fraction.

**Keywords:** image processing, snakes, genetic algorithms, cardiac MRI.

## 1 Introduction

The segmentation of the heart is a common task to be performed in cardiac exams of Magnetic Resonance Imaging (MRI). In one typical exam many images are obtained from different positions of the heart and at different phases of contraction (from systole to diastole). The segmentation is then performed off-line on these images to extract important clinical parameters and information that characterize the function and the anatomy of the heart. For instance, in order to calculate the cardiac ejection fraction or cardiac ejection curves, important parameters that characterize cardiac contraction [1], the medical specialist may need to segment near one hundred of two-dimensional images for a single patient. The cardiac ejection fraction is the relation of the blood cavity volume of the left ventricle during diastole per the volume during systole. For estimating each volume one segments the endocardium in different short-axis images, or slices, of the ventricle (around 10 slices from apex to base) and calculates the areas of the blood cavity in each of these images. The majority of today's commercial software provides segmentation in a semi-automatic way. Therefore, during the segmentation of the cardiac endocardial surface the specialist is

forced to pick around six points in the border between cardiac tissue and the blood cavity of each short-axis image.

The segmentation of the cardiac MRI images is of extreme importance but it is, today, a tedious and an error-prone task. In this work, we investigate the automatic segmentation of the endocardium in cardiac MRI images using the active contour technique named Snakes[2] method. This method is based on a parameterized closed curve. Internal and external forces change the shape of the curve adapting it towards the structure we seek. This curve should be deformed until it minimizes an energy functional determined by the information extracted from the image and by restrictions imposed to the curve, like elasticity and rigidity. In order to achieve better results we have adapted and tailored the Snakes method to the specific task of automatic segmentation of the endocardial surface of the left ventricle of the heart in MRI images. We propose and implement a new external force that further restricts the deformation and overcomes artifacts, such as the papillary muscles. The new method provided better results than those of the traditional Snakes method.

Nevertheless, the results of both implemented methods have shown to be very dependent on their configuration parameters. In this work, we use a Genetic Algorithm (GA) to find the best set of configuration parameters for the automatic segmentation method. The GA is used to solve a minimization problem, where the objective function captures the distance between 25 manually segmented short-axis images and the respective automatic segmented ones. We evaluated the results of the automatic algorithm using 150 manually segmented curves generated by a medical specialist using the best set of configuration parameters found by the GA.

As mentioned before, a single patient exam may consist of over a hundred of cardiac segmentations. Therefore, the computer implementations here described explore the embarrassingly data parallelism of the algorithms using Java threads on a shared-memory machine.

## 2 Snakes

In this model we consider a Snake as a parameterized curve such as:

$$\mathbf{v}(s) = [x(s), y(s)] \quad (1)$$

where  $x$  and  $y$  are functions which determine the coordinates of the function  $v$ ,  $v$  is the position of the snake and  $s$  varies from 0 to 1.

The main idea of the method lies in the minimization of an energy function that involves the Snake curve and features of the image [2, 3]. The energy  $E$  associated to the curve is defined in such a way its value is minimum when the curve is near the region of interest, i.e. near the borders of the image:

$$E = \frac{1}{2} \int ((\alpha \cdot |v'(s)|^2 + \beta \cdot |v''(s)|^2) + E_{ext}(v(s))) ds \quad (2)$$

The coefficients  $\alpha$  and  $\beta$  represent the elasticity and rigidity of the Snake, respectively, and they define its internal energy.  $E_{ext}$  is the energy term related to image characteristics and will be more discussed later. The operators  $|v'(s)|^2$  and  $|v''(s)|^2$  are the L2 norm of the first and second derivatives of  $v(s)$ , respectively.

The elasticity coefficient makes the Snake more resistant to traction. The rigidity coefficient makes it more resistant to bending. These two parameters prevent the Snake to become non-continuous or to break during the iteration process of the optimization problem.

The optimization problem above is solved using variational calculus, or being more specific, using the Euler-Lagrange equations[4]. A formal derivation of Eq. (2) gives us:

$$-2 \frac{d}{ds} \left( \alpha \frac{dv}{ds} \right) + 2 \frac{d^2}{ds^2} \left( \beta \frac{dv^2}{ds^2} \right) - \|\nabla I\|^2 = 0 \quad . \quad (3)$$

where  $\nabla I$  is the classical external energy and represents the image gradient.

The original external energy from the Snake model,  $E_{ext}$  is based on the image gradient,  $\nabla I$ , which points towards the image borders. The main problem of this approach is that it forces the initialization of the Snake to be very close to the region of interest. To overcome this problem we implemented the Gradient Vector Flow (GVF) method [3] with the purpose of increasing the region of influence of the external forces. This is obtained by solving a few steps of the diffusion equation (with a diffusion coefficient equals to  $\gamma$ ) to the components of the gradient vector. This procedure spreads the information of  $\nabla I$  all over the image [3, 5].

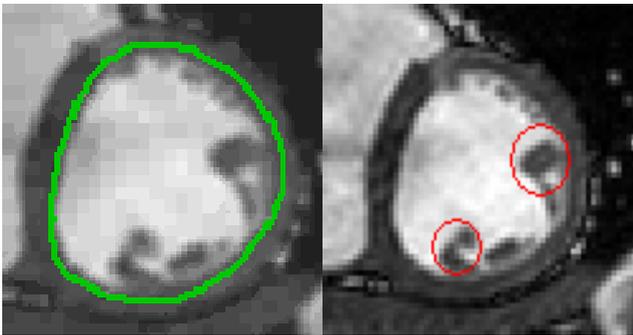
## 2.1 Balloon and Adaptive-Balloon Techniques

There is a particular problem during the segmentation of the endocardial surface: the existence of artifacts that have the same contrast of the object of interest. The presence of these artifacts should be ignored. This is the case of the papillary muscles highlighted in Figure 1.

To overcome this problem we have implemented the Snake method with the so called Balloon Force:

$$F_{Balloon} = k_l \cdot \vec{n} \quad . \quad (4)$$

In this expression  $k_l$  is the constant strength of the force and  $n$  is the unitary normal vector to the curve. Depending on the signal of  $k_l$ , the force will provide the curve to grow or to shrink. In our implementation it always grows.



**Fig. 1.** A correct segmentation (left) avoiding artifacts (right)

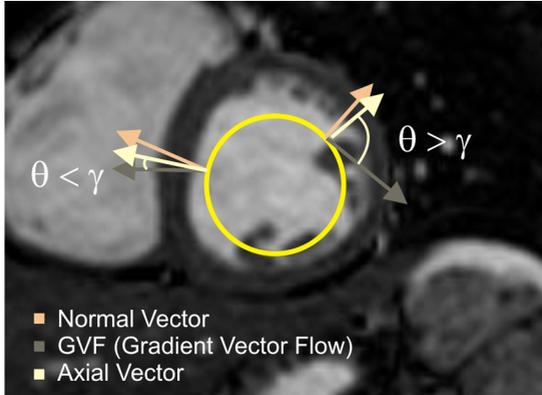
However, the balloon force has shown not always to work properly. It was very difficult to adjust the strength parameter. It should be strong enough to overcome the noise and the artifacts but not so strong as to overcome the borders of interest and grow disorderly.

To solve this problem we developed a new force named the Adaptive-Balloon. Its direction is also normal to the curve, but the intensity varies along the Snake and during the iterations of the method. The strength of this force depends on the information of the curve neighborhood.

Since the format of the object to be segmented is convex, we create a force that acts only over those Snake points that have angles between the vector of the GVF and the axial vector at  $v(s)$  greater than a predefined value. The axial vector of  $v(s)$  connects  $v(s)$  to the center of gravity of the image. The equation for the Adaptive-Balloon force is:

$$\begin{aligned} F_{\text{adapt}}(v(s)) &= k_2 \cdot \vec{n} \text{ , if } \theta > \gamma. \\ F_{\text{adapt}}(v(s)) &= 0, \text{ if } \theta < \gamma. \end{aligned} \quad (5)$$

where  $\theta$  is the angle between the axial vector and the GVF at the Snake point  $v(s)$ ,  $k_2$  is the force strength,  $\vec{n}$  is the unitary normal vector and  $\gamma$  is the limit angle previously chosen. Figure 2 shows some examples of the directions of GVF, the axial vector and the angle  $\theta$ .



**Fig. 2.** The calculation of the Adaptive-Balloon force based on the angle  $\theta$  between the GVF and the axial vectors

### 3 Genetic Algorithm

With the Adaptive-Balloon force, the automatic segmentation generated better results than those obtained by the Snake with the classical Balloon force. Nevertheless, the results of both implemented methods have shown to be very dependent on their configuration parameters. One set of parameters may work well in one image, but fail for another. It was not possible to manually find a single set of

parameters that produced satisfactory results for all images. Therefore we chose to use a Genetic Algorithm (GA)[6] to find the best set of configuration parameters for the automatic segmentation method. The set of parameters are: The elasticity  $\alpha$ ; rigidity  $\beta$ ; the viscosity  $\gamma$ ; the maximum amplitude associated to  $\nabla I$ ,  $\kappa$ ,  $k_1$  and  $k_2$ , the strength of the Balloon forces as in Eqs. (4) and (5), respectively.

The GA was used to solve a minimization problem, where the objective function captures the distance between 25 manually segmented short-axis images and the respective automatic segmented ones. For each curve we calculate the distance between the curve points and the center of gravity. The center of gravity is taken as the mean of the coordinates of the semi-automatic segmented curve. Therefore, the GA minimizes the error defined by the distance between these two curves:

$$\text{Error} = \frac{\sum_{i=0} (Rp_i - Rq_i)}{\sum_{i=0} Rp_i} \quad (6)$$

where  $Rp_i$  and  $Rq_i$  are distances from the center of gravity to the  $i^{\text{th}}$  point of the semi-automatic segmented curve and the automatic segmented curve, respectively.

The Genetic Algorithm simulates the evolution of a population of individuals, in our case, the Snake configuration parameters. The individuals, i.e. the Snake configuration parameters, were coded using real numbers. As the population evolves the best adapted individuals survive whereas the others die. The measure of the adaptation level of each individual is called fitness. In our case, the fitness measures the mean distance between the curves obtained automatically and the corresponding manual ones for 25 short-axis images, as described in Eq. (6).

To generate the offspring, the best adapted individuals are selected from the population. The Roulette-Wheel, also known as Fitness Proportionate Selection method [6], was used as the selection algorithm. After the parents are selected, they reproduce, generating the offspring. This process is called crossover. We implemented the Blend Crossover method [7] with a crossover rate of 90%. The offspring may mutate in the process. When this occurs, the new generated individuals are set with new randomly chosen genes within the range of each parameter. The mutation rate was set in a way it affects 10% of the population.

Half of the new individuals are selected to survive regardless of their fitness. The other half is made of individuals with higher fitness than their parents. This way, the individuals with higher fitness survive and we keep a diverse population. The method called elitism [6] was also implemented. It guarantees that the best individual of the population always survives after the selection process. The process described above was repeated two hundred times with a population of one hundred individuals.

## 4 Implementation

The algorithms developed were implemented in Java [8]. The Java language was chosen due to its rapid prototyping feature, its embedded support for concurrent programming and the wide availability of libraries for numerical methods and image

processing. In particular, the software was developed using the ImageJ framework [9]. ImageJ is a public domain software which offers many functionalities to image processing. One of its most interesting features is the possibility to extend its basic functionalities by the development of new plug-ins. That way, new plug-ins were created on ImageJ: two Snake methods, one with the classical Balloon force and another with the new Adaptive-Balloon force; and the Genetic Algorithm.

The implementation of the Snakes methods was based on the finite difference method. Each deformation of the Snake involves a linear system to be solved. The Cholesky method was implemented to solve the linear systems.

A MRI exam generates many images that need to be segmented. Therefore, the implemented algorithms work with sets of images. Each set of images is stored in an internal data structure of ImageJ called ImageStack. The ImageStack is processed in parallel. The parallel implementations explore the embarrassingly data parallelism and were developed for shared-memory machines using Java threads[10]. The implementation of the parallelism is conceptually simple. ImageJ loads the images from the ImageStack. The plug-in is executed and behaves as a master thread. The master creates new threads, the slaves. Each slave applies the Snakes method to a set of images of the stack and draws the segmentation results directly in the images. Figure 3 illustrates this process.

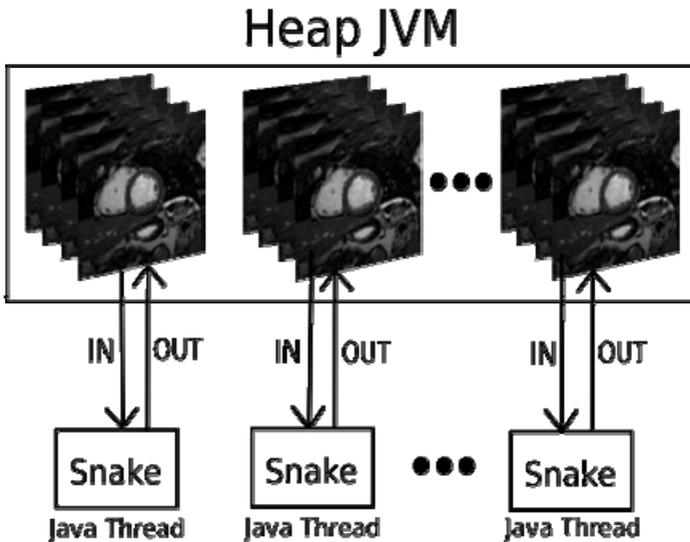
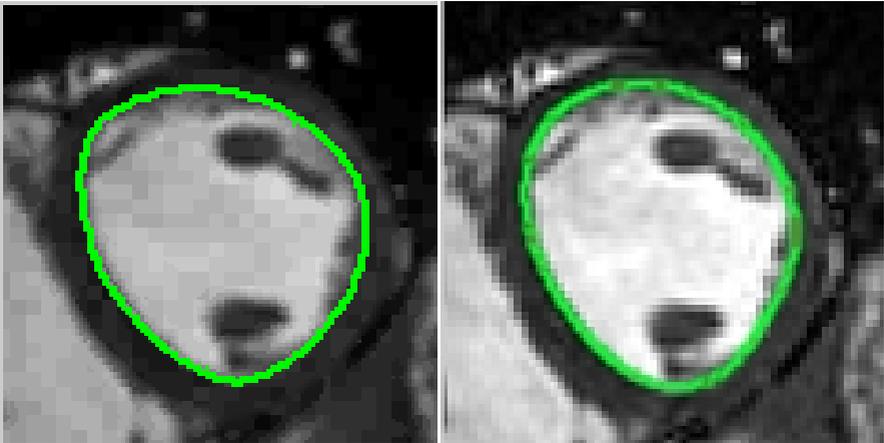


Fig. 3. The parallel implementation

## 5 Experimental Evaluation

In this section, we present experimental results obtained with a dual Xeon 1.6 GHz, 4 MB of cache, 4 GB of main memory. Each Xeon is a dual core processor. Therefore, all

computations were executed using 4 threads. The methods for automatic segmentation were tested using 6 stacks of images obtained from two healthy volunteers. There are 3 stacks per person, each from different short-axis position (apex, mid and base). Each stack has 25 images obtained from different phases of contractions, from diastole to systole and back to diastole. The parallel implementation achieved near linear speedups when using 4 threads. The automatic segmentation of 150 images took less than 5 seconds with 4 threads. To validate the methods the automatic segmentations were compared to manual ones performed by a medical specialist. Figure 4 presents the automatic and manual segmentations for one of the images.



**Fig. 4.** Manual (left) and automatic segmentation using Snake-Adaptive Balloon method (right)

**Table 1.** Mean errors using the automatic methods

	Balloon	Adaptive-Balloon
Stack 1	12.2%	10.4%
Stack 2	9.6%	8.3%
Stack 3	9.4%	6.9%
Stack 4	13.4%	11.9%
Stack 5	9.9%	9.1%
Stack 6	9.7%	7.9%
<i>Average</i>	<i>10.7%</i>	<i>9.0%</i>

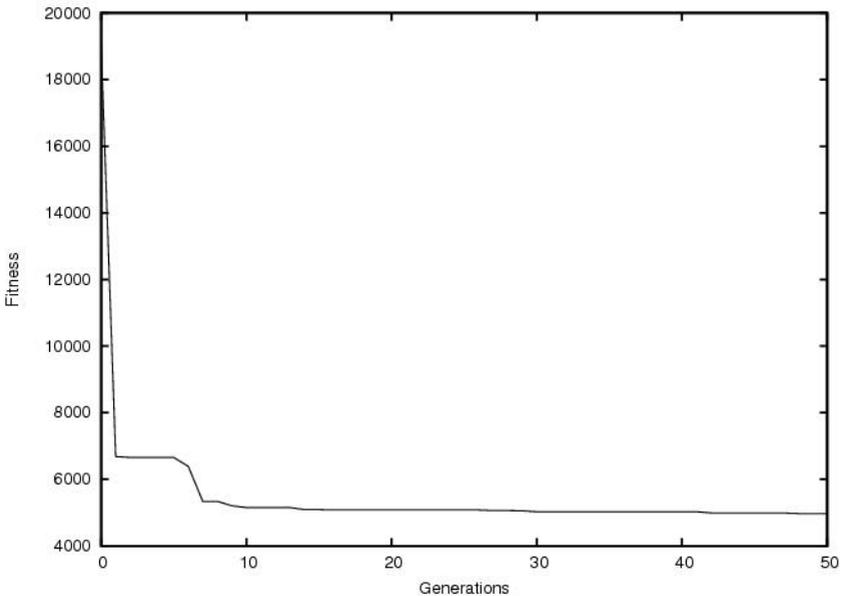
### 5.1 Adaptive-Balloon vs. Balloon Force

As we mentioned before the tuning of parameters for the Snakes methods is a non-trivial task. A set of parameters may achieve good results for one image but not for others. To compare the new developed Adaptive-Balloon force to the classical one,

we have first manually tuned the parameters using the Snakes method with the Balloon force for each image. After that, using the same sets of parameters, we executed the Snakes method with the extra Adaptive-Balloon force. Table 1 presents the errors of segmentations obtained with the automatic method compared to the manual segmentations. The error presented in this table is the distance between the two curves, as defined by Eq. (6). The table shows that the new Adaptive-Balloon force technique achieved better results than the classical one.

## 5.2 Validation of the Method

The Genetic Algorithm optimized the parameters using one of the stacks of 25 images. Figure 5 shows the evolution of the fitness, i.e. the error as defined in Eq. (6), of the best individual (parameter set) per generation. After a few generations we observed that the fitness drops an order of magnitude.



**Fig. 5.** Fitness value per generations

**Table 2.** Errors between automatic and manual segmentations

	Mean Error	Standard Deviation
Stack 1	9.8%	3.8%
Stack 2	7.0%	1.8%
Stack 3	6.3%	1.8%
Stack 4	10.6%	3.5%
Stack 5	7.6%	2.7%
Stack 6	7.3%	2.5%
<b>Average</b>	<b>8.1%</b>	<b>2.6%</b>

The best set of configuration parameters found by the Genetic Algorithm was used to evaluate the effectiveness of our Adaptive-Balloon force technique. Table 2 presents the segmentation errors obtained when applying these configuration parameters to the stacks. Again, the error corresponds to that of Eq. (6). As we can observe, the proposed method obtained satisfactory results, with errors around 8%.

## 6 Conclusion

The segmentation of the cardiac MRI images is of extreme importance but a tedious and an error-prone task, whereas the automatic segmentation is a challenging task. In this work we proposed, implemented and evaluated a new external force technique, called Adaptive-Balloon, to improve the convergence of the Snakes method, traditionally used to segment images. In addition, a genetic algorithm is used to find the best set of configuration parameters for the Snakes method. The proposed method obtained satisfactory results, with errors around 8%. We developed a parallel implementation to explore the embarrassingly data parallelism. The parallel implementation was very effective, segmenting automatically 150 images in less than 5 seconds. The preliminary results suggest that the methods are promising and with further development and validation they may be used, for instance, for the automatic calculation of cardiac ejection fractions.

## Acknowledgments

The authors would like to acknowledge the reviewers for their comments and suggestions that significantly improved the quality of this work. This work was supported by CNPq, FAPEMIG and UFJF. G. M. Teixeira is a scholarship holder of PIBIC/CNPq.

## References

1. Kühl, H.P., Schreckenberg, M., Rulands, D., Katoh, M., Schäfer, W., Schummers, G., Bücken, A., Hanrath, P., Franke, A.: High-resolution Transthoracic Real-Time Three-Dimensional Echocardiography: Quantitation of Cardiac Volumes and Function Using Semi-Automatic Border Detection and Comparison with Cardiac Magnetic Resonance Imaging. *J. Am. Coll. Cardiol.* 43, 2083–2090 (2004)
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. *International Journal of Computer Vision* V1(4), 321–331 (1987)
3. Xu, C., Prince, J.L.: Gradient Vector Flow: A New External Force for Snakes. In: *IEEE Proc. of the Conference on Computer Vision and Pattern Recognition*, pp. 66–71. IEEE Computer Society, Washington (1997)
4. Forsyth, A.R.: *Calculus of Variations*. Dover, New York (1960)

5. Xu, C., Prince, J.L.: Generalized Gradient Vector Flow External Forces for Active Contours. *Signal Processing* 71(2), 131–139 (1998)
6. Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*. Springer, Heidelberg (2003)
7. Eshelman, L.J., Schaffer, J.D.: Real-coded Genetic Algorithms and Interval-Schemata. In: *Foundations of Genetic Algorithms-2*, pp. 187–202. Morgan Kaufman Publishers, San Mateo (1993)
8. Arnold, K., Gosling, J., Holmes, D.: *The Java Programming Language*, 4th edn. Prentice Hall PTR, Englewood Cliffs (2005)
9. ImageJ, <http://rsb.info.nih.gov/ij/>
10. Doug, L.: *Concurrent Programming in Java: Design Principles and Pattern*, 2nd edn. Prentice Hall PTR, Englewood Cliffs (1999)