2019

# Trapping ACO applied to MRI of the Heart

Shannon Lloyd Birchell
*University of North Florida*

# Trapping ACO applied to MRI of the Heart

by

Shannon L. Birchell

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computing and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

April, 2019

The thesis “Trapping ACO applied to MRI of the Heart” submitted by Shannon L. Birchell in partial fulfillment of the requirements for the degree of Master of Science in Computing and Software Engineering has been:

Approved by the thesis committee: Date

_____ _____

Dr. Sherif Elfayoumy
Thesis Advisor and Committee Chairperson

_____ _____

Dr. Roger Eggen

_____ _____

Dr. Sanjay Ahuja

Accepted for the School of Computing:

_____ _____

Dr. Sherif Elfayoumy
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

_____ _____

Dr. William Klostermeyer
Interim Dean of the College

Accepted for the University:

_____ _____

Dr. John Kantner
Dean of the Graduate School

Acknowledgements

First, I would like to thank the many people around me that have provided persistent support throughout my academic endeavors. It has been a turbulent experience returning to school in midlife, in the end, it has transformed me into a better more contributing person in our diverse society. I would like to thank some of the professors that have taken a personal passion in education and vested interest in their student's. None-inclusive non-ordered list that I have been personally exposed to: Dr. Eggen, Dr. Eflayoumy, Dr. Behrooz Seyed Abbassi and Dr. Chuan. A special thanks to Dr. Eggen and Dr. Elfayoumy for working with me during my busy career. Without their assistance and understanding this would not be possible. Lastly, I would like to thank my sons for the strength and love.

CONTENTS

FIGURES

TABLES

EQUATIONS

# ABSTRACT

The research presented here supports the ongoing need for automatic heart volume calculation through the identification of the left and right ventricles in MRI images. The need for automated heart volume calculation stems from the amount of time it takes to manually processes MRI images and required esoteric skill set. There are several methods for region detection such as Deep Neural Networks, Support Vector Machines and Ant Colony Optimization. In this research Ant Colony Optimization (ACO) will be the method of choice due to its efficiency and flexibility. There are many types of ACO algorithms using a variety of heuristics that provide advantages in different environments and knowledge domains. All ACO algorithms share a foundational attribute, a heuristic that acts in conjunction with pheromones. These heuristics can work in various ways, such as dictating dispersion or the interpretation of pheromones. In this research a novel heuristic to disperse and act on pheromone is presented. Further, ants are applied to more general problem than the normal objective of finding edges, highly qualified region detection. The reliable application of heuristic oriented algorithms is difficult in a diverse environment. Although the problem space here is limited to MRI images of the heart, there are significant difference among them: the topology of the heart is different by patient, the angle of the scans changes and the location of the heart is not known. A thorough experiment is conducted to support algorithm efficacy using randomized sampling with human subjects. It will be shown during the analysis the algorithm has both prediction power and robustness.

# Chapter 1

## Introduction

Heart disease is a ubiquitous problem causing more than 610,000 deaths in the United States alone [NCHS15]. Cardiomyopathy is one of several heart ailments that arises from inability to displace blood. Patience's are routinely given MRI's for diagnostics purposes to help establish the aliments and the extent of impact on their health. MRI's provides information by taking slices of high definition images through the body at various orientations and locations. It is possible to calculate volume of blood through many MRI slices of the heart with the laborious efforts of a professional to identify the heart boundaries. Only after the doctor has outlined the boundaries of the heart through a complete heartbeat cycle, can the calculation the displaced blood volume be performed.

The process of manually circumscribing the blood in the heart is very inefficient and error prone. Adding to the doctor's significant efforts, circumscribing must be performed on both left and right ventricle over many MRIs. This is a mundane and relentless task since there are k*n MRIs to analyze (where k is the number of slices and n is the time count), which equates to normally more than a 100 MRIs. Circumscription can optimistically take approximately 30 seconds per image, resulting in a conservative estimate of 50 minutes for each patient.  Due to the substantial remuneration of doctors, this implies significant cost to insurances providers and clients.

In addition, manual drawing of heart edges can become somewhat a subjective process resulting in identification inconsistencies. This subjective nature is a result of the MRI themselves and not so much about the doctors' capability or skill set. Artifacts such as spots, noise and textures can obscure the boundaries and require interpolation. The interpolation may not be consistent between professionals leading to improper metrics.

It is possible to automate the laborious task of circumscribing using known methods of optimization. This can save tremendous amounts of time enabling professionals to increase productivity while reducing or eliminating subjectivity. There are a multitude of optimization techniques, of which Ant Colony Optimization (ACO) is included. ACO is a model that attempts to simulate real world ants guided by a goal. We use these theoretical ants by placing them on MRI's and setting the goal as detecting heart boundaries. In this research we will elaborate and hopefully establish solidarity that ACO can adequately identify heart walls and thus, implicitly, support heart volume displacement calculations.

## 1.1 Background

This research relies heavily on image segmentation and ant colony optimization. Regarding image processing there are a few basic concepts that are needed. An image is made of pixels that are associated with a color. In this research monochrome images (grayscale) are used which results in a single integer value to representational the intensity. This value determines the lightness or darkness of each pixel. All images are

two-dimensional resulting in every pixel having four neighbors (except image edges). Relative to some pixel, there is one path of length one to each of the neighboring pixels.

Ant colony optimization is a heuristic based on actual ant colonies [Dorigo04]. Ants communicate using Stigmergy, the process of changing the environment to communicate with other intelligence agents (ants). Stigmergy is done though dispersion of pheromone on the path of travel. When ants encounter pheromone, they are more susceptible to go in the direction of the pheromone. The assumption is that many trials will eventually cause the ants to converge to an optimal solution.

Image segmentation is the process of converting an image from multi-valued pixels to, in regard to this research, dichotomized binary values. There must be a value chosen prior to the segmentation called the threshold. After this value is chosen the image is simply converted as follows:

If pixel value is greater than threshold change to color A

If pixel value is lower or equal than threshold change to color B

A useful image segmentation rests entirely on the value of the threshold making it very important that it is selected with care. A simple way to select a threshold is to take the average of every pixel value in the image. Although this may work in some situations, there is normally more sophisticated methods used such as receiver operation characteristic [Vermon91].

### 1.1.1 MRI

Heart MRI's have a somewhat consistent topology and generally look like Figure 1. As previously discussed, there can be different orientations and locations resulting in different views of the heart. The current image was chosen as it contains several characteristics while still maintaining a clean descriptive view of the heart. An expression of areas that look consistent are generally referred to as the salient region. When using the phrase "salient region" here it is meant as a loosely identified homogenous area. Referring to the image we see several possible salient regions; three predominate salient regions have been identified as the heart (labeled 1 and 2) and lung (labeled 3). It is important to note many salient regions can be devised in the image and may not be useful or even biologically logical. These different possible salient regions can cause issues when detecting the heart. Intra-salient region attributes can have different intensities, gradient contours and textures which increasing the complexities of the problem space. The quintessential difference between interesting and uninteresting salient regions is contrast, size and location.

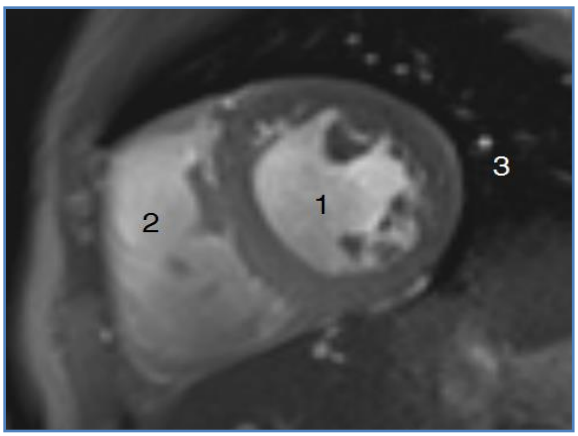


Figure 1: MRI Heart Regions

Referring to the image, the blood in the heart has significantly lighter intensity than other regions, such as the lungs. The heart wall has a relatively darker intensity than the blood but not as dark as the lungs. This implies a good segmentation algorithm may identify the heart walls but, this is not the case. When thresholding the dichotomy is not discriminatory towards the heart wall and many other artifacts may be confounded. Further, as discussed, segmentation relies on a threshold at a precise value which will not be optimal for different images [Kaman10]. Heart walls usually do not have a consistent intensity leaving the segmentation method lacking. Looking at region 1, there is a wide range of pixel intensities within the heart boundary. These variations within the salient regions are not part of the heart and should be suppressed when detecting boundaries.

For the most part there are abrupt changes from salient regions to tissue, but this is not consistently a valid assumption. As shown in the lower part of salient region labeled 2, the salient region and boundary are not readily discernable. Therefore, edge detection

such as Sobel [Sobel14] will find some of the heart boundaries but will fail at critical points. This is an unacceptable behavior as interpolation of missing points may cause serious discrepancies in heart volume. Still referring to region labeled 2, inside the salient area one can make out additional lines that are not part of the boundaries of the heart. This is due to the blood pressure variances, resulting in different densities, under normal heart function. Therefore, lines do not necessitate a meaningful boundary when detecting ventricles.

MRIs of the heart do provide quality information but have many difficulties arising from complex characteristics. These complexities cause both professionals and technological solutions to have opportunities. Further it has been established here that detections algorithms based on edge detection or segmentation cannot provide a panacea to automation of volume calculation.

### 1.1.2 Watershed

The Watershed model is a topological algorithm based on water aggregations. This is done by transforming a gray scale image into three dimensions. The third dimension is the intensity of the pixel and is interpreted as elevation. Therefor if water is evenly distributed throughout an image it will flow with the gradient and aggregate at all local minima. Each of the distinct aggregations are identified as an independent cell. These cells are completely separately from each other by pixel intensity or elevation. If water is

continuously added, regions will begin to merge as the elevation thresholds are surpassed. One good aspect of this methodology is that local segmentation thresholding is implicitly performed.

Normally, before image is processed the gradient transformation is performed. This "normalizes" the image and exaggerates the edges since, by definition, larger differences exist in neighboring pixels on edges. This implies that as threshold increases more distinct edges are detected since the derivative will be greater. An alternative to gradient can be used called the lower neighborhood (LN) [Masoumi12]. The use of LN reduces process requirements since it is simply the lowest value of neighborhood. The LN is a non-parametric method that decreases complexity and stabilizing outliers.

The algorithm divides the image into local minima cells by manual marking areas or marking positions with gradient < k as a local minimum. To simulate region flooding, the local minima are incrementally increased by changing the starting threshold intensity value k by l. This results in contiguous pixels that are within the new value threshold to be added to the associated cell set. In Figure 2, cells may flood horizontally or vertically. The starting position is the center, setting k equal to 50, the pixels notated by light blue "fill up" or merge.

| | | | | |
|---|---|---|---|---|
| 80 | 80 | 80 | 80 | 80 |
| 80 | 66 | 54 | 50 | 80 |
| 80 | 53 | 49 | 53 | 80 |
| 80 | 48 | 50 | 71 | 80 |
| 80 | 59 | 54 | 65 | 80 |
| 80 | 54 | 56 | 63 | 80 |
| 80 | 80 | 80 | 80 | 80 |

Figure 2: Watershed

If 5 is added, resulting k = 55, the boundary is breached resulting in the dark blue areas becoming flooded.

The example provides insight to how cells are generated as the water level is increased. Cells are the resulting states through which k changes and can be represented as a graph. Let there be graph G(V,E) such that V is the set of vertices and E is the set of weighted edges. If water fills up a region of pixels, it must have traveled a path to each pixel. Also, the path traveled is weighed since some pixels have lower values causing more attraction. Then it can be shown if G' is a connect subgraph of G and it is the minimal spanning set, then it is equivalent to watershed method. Succinctly, let G' be a subgraph of G then the minimum spanning set is equivalent to the watershed [Jean-Yves10]. What is meant by state can now be posed in the right context. State is processed through several iterations and uses the graph theory concept of component extension. Let component $G'_t$ be a

component of G at time t then at time t+1, $G'_{t+1} \subseteq G'_t$. This implies that $G'_t$ is an extension of $G'_{t+1}$ and since threshold t is either monotonically increasing or decreasing then $G'_t$ is either monotonically increasing or decreasing, respectfully.

The application of Watershed methodology on most images results in a significant number of cells, refer to the second image in Figure 3. This is undesirable and results in a poor solution or substantial post processing. Therefore, prepossessing the image with a various method before performing this algorithm becomes a necessity. Another major difficulty is establishing the threshold value that extracts the desired results. If this value is ill defined, there may be substantial merging of cells or too little merging. To combat these downfalls other techniques are used in conjunction with Watershed. In Figure 3 , the original MRI of the liver is on the left, extreme number of cells in the center and too few cells on the right.

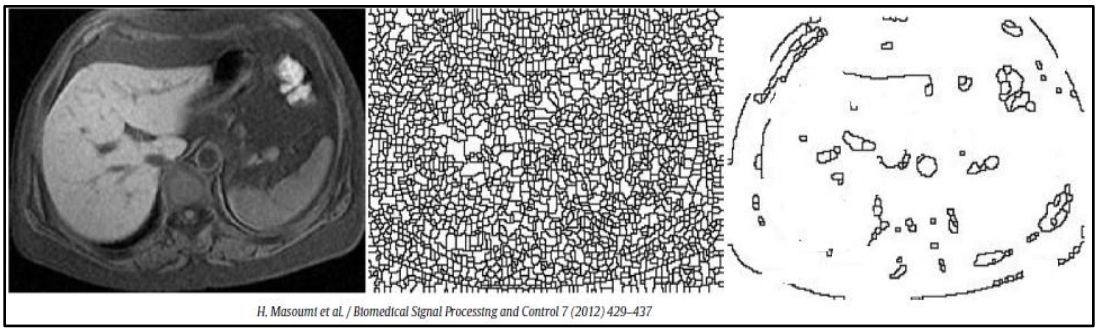



Figure 3 Liver Watershed

Conditioning can be used to assist the Watershed method. One type of preprocessed ubiquitous throughout image processing is the application of Gaussian filters. This retracts differences in outlier intensities through averaging. Gaussian filtering is accomplish using a sliding window of size n x n applied to every pixel in the image. The values of each of the windows are calculated and the center pixel value replaced with the new values. The weighted value is calculated using the Mahalanobis distance with the Gaussian distribution as the coefficient. The new value of p is $p_{x,y} = \beta_1 p_{d=1} + \beta_2 p_{d=2} \ldots \beta_n p_n$ were $\beta_1 + \beta_{2 \ldots} \beta_n = 1$ and d is distance. In Figure 4 images display the effects of a Gaussian filter applied iteratively. The images sequenced from left to right, displays a blurring until there is not enough information to discern the person.


Figure 4: Gaussian Filter

The application of Gaussian transformation operator, or convolution, results in a Morphological Smoothed Image (MSI) which reduces intensity extremes and noise such as speckle. Taking convolutions further, it is possible to use mixed Gaussian masks. This is done by using windows of different sizes and assigning pixels the difference of the

masks. Further these methods have been combined with morphological operators and applied to MRI image processing. The morphological closure and gradient operators are used in conjunction with a Gaussian mask for liver detection [Masoumi12].

There are several methods to determine threshold values to create cells. A good threshold values may change between images and thus one value is not a fit all. Therefore, techniques must be used to find this value. One simple technique queries the image to set t = (max (p) – min (p)) /2 [Zhao08]. Although this provides an approximation of central tendency, in most cases, this is a naive guess. It is incorrect to assume that the edges of interest require a centralized threshold. Still, a more sophisticated technique is required.

A concept that is successful in establishing good threshold values is iterative processes based on optimization. This can come in many forms but at its most general level, consists of trials and a fitness function. As trials increase the fitness function works to minimize error based on threshold perturbation.

One technique used is the iterative Watershed with scaling based on Neural Network error detection [Masoumi12]. Supplied with an expected result the neural networks can establish the difference in shape and area against a paragon, difference in these characteristics result in an error value. This feedback mechanism injects thresholds values parameterizes watershed model. On the unset, threshold is set to max intensity then incremented lower until the NN tolerance becomes within a user defined specification. This scaling feedback method provided a good mechanism for optimization. One issue in

this feedback approach is the neural network must be trained to look for shapes and area. This leads to problems when shapes are morphologically inconsistent resulting in high error or no convergence.

Continuing to combat some of the downfalls of watershed, two more methods are briefly discussed called Hopfield neural networks, HNN [Storkey99] and Canny edge detection [Nadernejad08]. HNN is a recurrent network based on energy levels, resulting output is a binary value. Energy levels are altered equivalently to previously discussed thresholds in watershed. The neighborhood of each pixel provides the input that excites HNN to produce an output assigned to the center pixel as edge or no edge. This results in a watershed methodology guided by HNN. The HNN suffers similar setbacks as the core watershed method in that many cells can be generated diluting important information.

Another factor to consider is with the Watershed edges themselves. It is not desirable to have edges that contain spurs or are thick. To reduce thick edges the concept of "destructive" edges is used [Jean-Yves10]. Any pixel on an edge that is more than two connected is removed in an iterative process until done. As shown in Figure 5, the gray shaded circles are the original watershed output and the black shaded circle are the result after performing the destructive operation.

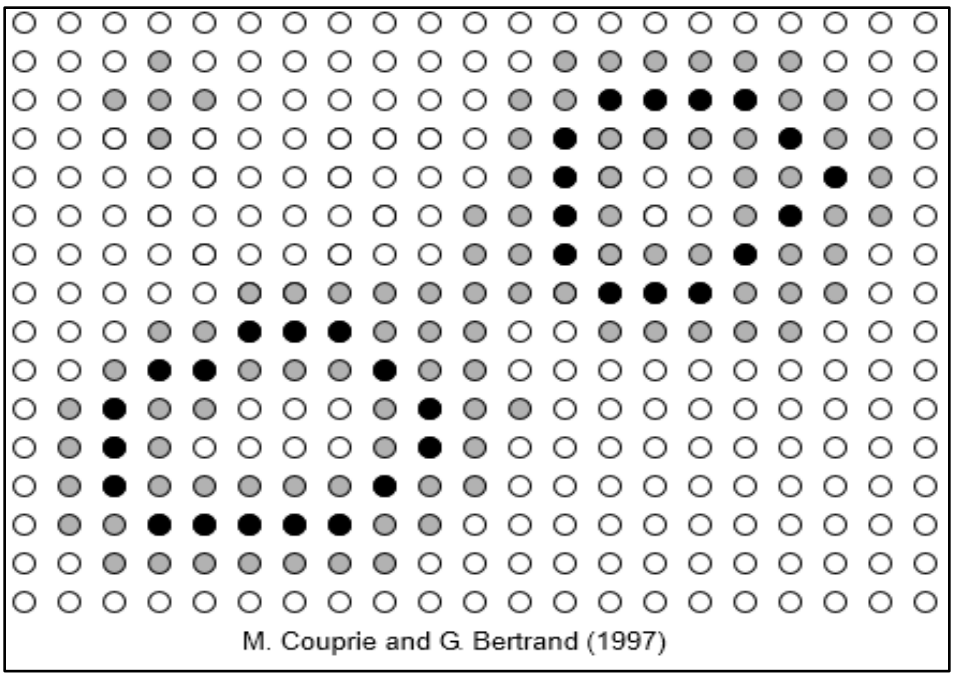



Figure 5: Watershed destructive edges

### 1.1.3 ACO

As discussed in the introduction, ant colony optimization is a model based on real ants. Ants disperse pheromone in the environment as a form of information which can be acted on by other ants in their forging process. There are several different interesting variants of this model each having different advantages. Some ACO variants help to converge a solution faster while others may provide more robust solutions at the expense of slower convergence. Well documented ACO algorithms such as stubborn ants [Abdelbar03], Elitist ants [Abdelbar08], Min Max ants [Stutzle00] and threshold ACO [Zhao08] have showed promise. To understand some techniques several are subsequently surveyed.

### 1.1.4 Standard ACO

To understand the many ACO we will review, arguably, a standard form as applied to image analysis. An ACO supplied with heuristics, or a utility function, has the ability to make decisions based on pixel intensities. An image can be mapped into a perceptual graph with nodes and edges [Zhuang04]. The representation binds nodes to pixels and edges to paths, where a neighborhood of nodes is all nodes one path away from a refence node. Referring to Figure 6, the circles represent the nodes, or pixels, and the lines between the pixels represent paths. Valid paths are up, down, right and left but not diagonal as shown with the circle containing the x. In the figure, "pxl 1" represents an edge pixel and does path "up" does not exist. Referring to "pxl 2", it is shown that four paths exist where each path has a weight as: 3.5, 2.6, 1.1, 2.1. Also take note that "pxl 1" is a neighbor of "pxl 2" as well as the converse.

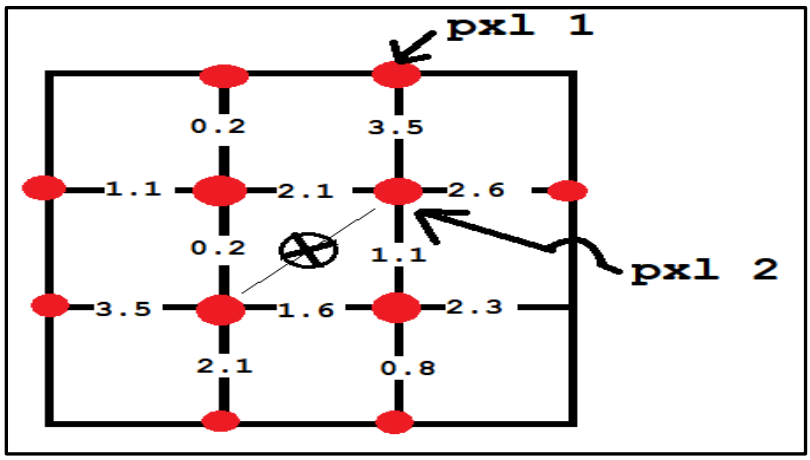



Figure 6: Image representation for ACO

The weights between the pixels, or on the path, are the driving influencer of the ants routing choices. As a neighbor's weights relatively increase in the neighborhood it becomes a more compelling choice. These weights and actions are derived by random proportional transition rule [Monteiro09] that takes input as input the neighborhood of nodes and edges. The path weights are updated recursively throughout the forging process, or many trials.

Referring to the top formula in Equation 1, let the probability of following a path to pixel i,j be $P_{ij}$ for trial number t. Let $\tau_{ij}(t)$ be pheromone density and $\eta_{ij}(t)$ be the change in pixel intensity [Zhao08] on each path in the neighborhood. The user defined values of $\alpha$ and $\beta$ foster ant behaviors such as randomness and pheromone bias. As $\alpha$ increases ants will follow past paths whereas increasing $\beta$, behavior will change to favor image attributes. The denominator of the formula is the sum of all neighbor values providing a normalized output suitable for probability. The numerator provides the bias or the weight of a specific path.

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t)$$

Equation 1: Standard ACO Formulas

In the second formula (η), this weight is inversely proportion to $d_{ij}$, . There can be a multitudinous of functions for $d_{ij}$ such as difference, max or variance of pixels, etc. The last formula t controls dispersion and dissipation of pheromone for trail location i j. Where p is the dissipation rate and $\tau_{ij}$ is a recurrent equation aggregating dispersion and dissipation for all trials.

Using this model as the gold standard, the following sections will explore variations and augmentations.

### 1.1.5 Stubborn Ants

Stubborn ants are compelled to follow their past paths to certain degree. To do this the ants are endowed with the capability to distinguish their pheromone from others [Abdelbar08]. Although the ant is attracted to pheromones its agency will put a high priority on its own pheromone. The intended benefit of this technique is to thoroughly evaluate regions, or slow convergence, causing an increased robustness of solutions.

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta} \cdot \delta_{i,j}^{k}(t-1)}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta} \cdot \delta_{i,l}^{k}(t-1)} \quad ,$$

$$\begin{aligned} \delta_{a,b}^{k}(t) &= \gamma \quad \text{if } (a,b) \in \mathcal{E}_{t-1}^{k} \text{ or } (b,a) \in \mathcal{E}_{t-1}^{k} \ , \\ &= 1 \quad \text{otherwise} \ , \end{aligned}$$

Equation 2: Stubborn at Formula

To augment the discussed standard ACO formula and implement a “stubborn” attribute, the updated formula shown in Equation 2 is shown as identified in Abdelba’s work. In the top equation, the standard formula has been outfitted with additional parameter δ and is normalized over the neighborhood. Reviewing the bottom formula, parameter δ is set to one when a past path is not encountered resulting in no bias. In the other case, δ uses γ as an additional heuristic contributing to bias, where nodes a and b belong to the previous edge at time t-1 traveled by ant k. For γ values increasing greater than one the bias becomes more prevalent. One issue with this approach is unsuccessful paths may be persisted since there is no way to gauge the fitness level. In the next algorithm, not only do paths persist like stubborn ants, they are acted on only when a certain level of fitness is achieved.

### 1.1.6 Elitist Ants

Elitist ants behave uniquely in that the pheromone dispersion depends on a utility function called Cost Based Abduction [Abdelbar03]. If an ant satisfies the utility function better than another arbitrary ant, it disperses pheromone that is more “attractive” than the arbitrary ant. Therefore, ants that perform better have more influence of the successive trails. Elitist model has variants such as k best ants, only the best ant or all ants may update but in a weighted configuration.

To establish if an ant has a better solution than its siblings and predecessors, it is first assumed there are an infinite set of hypotheses that are applied to cost-based abduction methodology. To promote economy, discussion is limited to the most import formulae in this report which may be refenced in Abdelbar & Mokhtar [Abdelbar03] if deeper understanding is required. The abduction formula is shown in Equation 3, where $X_{ih}$ is the hypothesis h of ant i establishing the probability Pr of the hypothesis being true.

An example hypothesis may be the most pixel differences of a path on some trial. Then paths with a greater summed difference in intensities would be a better hypothesis than paths with less. In this context the reduction of the cost function is the optimization method, therefor it is modified to be the inverse of the sum of pixel intensity. Looking at

the bottom formula in Equation 3, g is a weighted result of τ and η where η is a domain dependent characteristic and τ is the best solution or hypothesis. As the value of $\alpha$ is increased, the elitist pheromone will be more strongly encouraged from another pheromone.

Throughout trials the best solution called "Global Best" is saved and updated, Xg. This formula is show in Equation 4. As shown, if current iteration j has reduced the cost, function f, more than the global g, then it is selected as the new global.

$$Pr(x_{ih} \rightarrow true) = \frac{g(h, true)}{g(h, true) + g(h, false)} .$$

$$g(h, r) = [\tau_{hr}]^{\alpha} \cdot [\eta_{hr}]^{\beta} ,$$

Equation 3: CBA formula

$$x_g = \begin{cases} x_j & \text{if } f(x_j) < f(x_g) \\ x_g & \text{otherwise.} \end{cases} ,$$

Equation 4: Global best update

In the bottom formula of Equation 3 the two components are expanded in Equation 5. The top formula of Equation 5 transforms the pheromone values to be $\tau_{max}$, $\tau_{min}$ or $\tau_{r\,j}$

[Stutzle00]. The bottom equation is the heuristic as show previously in the standard model.

$$\tau_{hr} = \begin{cases} \tau_{max} & \text{if } \tau_{hr} > \tau_{max} \\ \tau_{min} & \text{if } \tau_{hr} < \tau_{min} \\ \tau_{hr} & \text{otherwise} \end{cases}$$

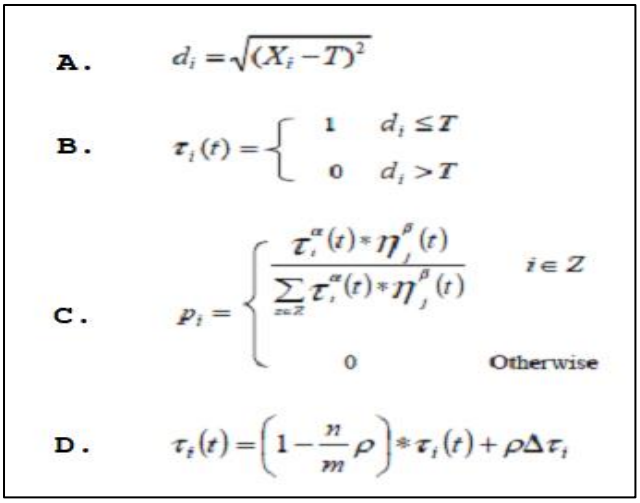
$$\eta_{hr} = \begin{cases} z(h) & \text{if } r = true \\ 1 - z(h) & \text{if } r = false \end{cases}$$

Equation 5: Elitist Update

This results in the standard ACO with a path that globally updates when it is found to be better

### 1.1.7 Thresholding with ACO

The last model discussed is a novel thresholding technique using ACO (TACO) [Zhao08]. Unlike the previous methods, every pixel is initialized with an ant. The ants perform all traversals on one pixel, although this seems contrary to ACO, the similarity holds in the selection of pheromone updating. TACO primary function is to dictate if an ant is “merged” into the solution. Merging is the processes of including pixels into the solution set for the best contrast.




Equation 6: TACO

Selection of initial "food" is set as the value of T= (max pixel value - min pixel value) / 2. In Equation 6 A, the distance of each ant is calculated to its corresponding pseudo pixel (itself), where X is pixel intensity and T the threshold value. The pheromone update is then performed using a binary function where distance is less than or equal to T set to 1 else 0 (Equation 6 part B). Finally, pheromone update is done in the bottom equation where p (Equation 6 D) As processing termination occurs under the constraint $|T_{curr} - T_{last}| < \epsilon$ where $\epsilon$ is a user defined variable.

## 1.2 Motivation

There are many variations to ACO that are used in various problems and medians. In previous discussion it was pointed out that ACO has been used in image analysis with optimistic results. Also discussed previously was the need to calculate blood

displacement of the human heart to detect aliments. Experiments where performed using ACO in the next section and a pilot algorithm in the after.

### 1.2.1 ACO and Watershed Applied to the Heart

In this section preliminary analysis is done to identify and understand characteristics of using ACO with MRIs. In Figure 7, there are three regions: 1 & 2 are both ventricles of the heart and 3 the lungs.


Figure 7: Transverse slice of hear

In Figure 8, a watershed was performed with 50 different levels. This means, metaphorically, 50 levels of water where poured on the image and basin perimeters recorded. As shown in the images, a plethora of salient regions are found but the regions of interest (1 & 2) are difficult or impossible to discern. A boon of the identified regions

is that they are internally consistent and can be rightfully assumed to belong to the same object.

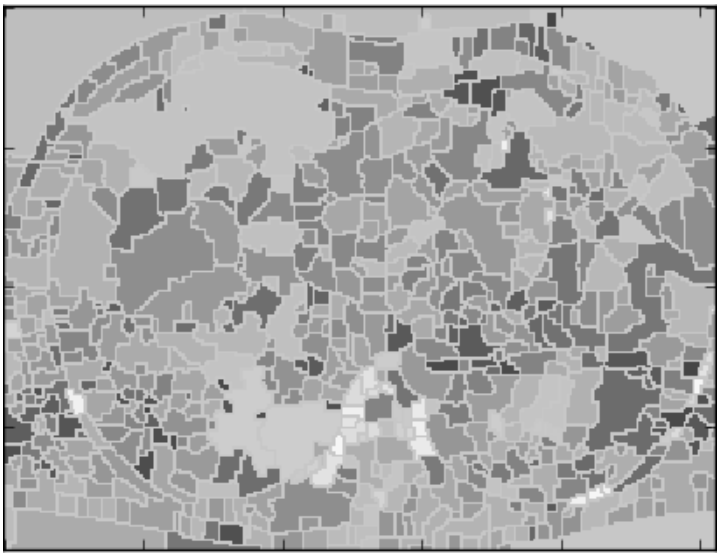


Figure 8: Watershed

The previously discussed method TACO is applied on the image, and results shown in Figure 9. The algorithm performed well in the sense of identifying regions. It is readily evident that lungs and heart are the white regions and black is the denser regions. Also, the edges are consistent and not over porous which means the value of the threshold selected was close to optimal choice. One serious issue is that region 1 is merged with region 3. This is a serious issue and is a downfall of contrasting in general. The merged salient region is caused by the boundary changing intensity to just below the threshold. If the threshold value is adjusted to allow for this ambiguity, then the salient regions become muddled and not useful. If regions are merged volume cannot be reliably calculated causing this method not to work.

Figure 9: Contrasting

The results in Figure 10 represents the best results from stubborn ants and elitist ants (with 40 trials), in this case, stubborn ants. The edges are unequivocally detected but have several hindering attributes. The edges are extremely thick causing two or more edges to fuse together. The edges are not smooth but jagged and harsh causing difficulties in establishing the actual heart walls. There are many artifacts that have nothing in common with the edges. The main causes of this seem to be that the resolution (number of pixels) of the image and transient edges. If the resolution is increased the time of processing increases significantly and may not be feasible. If assumed all edges are detected within some predefined standards, there would still be a significant problem determining which edges are heart walls. Therefore, every edge detection algorithm, shown here, does not discriminate from the edges that are of interest.

Figure 10: Edge Detection

Previous analysis provides jagged edges and merged region, although, the result was an accurate depiction of the original MRI image. A major issue, algorithms do not differentiate edges of importance such as heart walls and lungs. These algorithms require many trials to adequately find an optimized threshold values or the intensity differences. One main issue is the edges are not quantitatively defined thus cannot be directly translated for volume derivation. It is shown that the Left ventricle does not possess a closed region and therefore the edge was not completely detected. This would completely invalid volume calculation making this algorithm undesirable.

On the other hand, both segmentation and stubborn ants provided good information, although different information. The next section introduces a unique algorithm using concepts from both approaches.

### 1.2.2 Trapping Ant Algorithm (TAA)

The previously discussed algorithms all have benefits and opportunities. In this research an augmentation and merging of methods will be presented that results in elimination said problems while keeping the best attributes. To lay the foundation of the final algorithm an intermediate method was devised as the foundation and to prove the concept.

The Trapping Ant Algorithm (TAA) can be understood by imagining a small confined flat surface as the environment, then inserting a single ant into this area as an agent. The ant will then begin walking straight (approximately) until it finds an edge. When it is unable to continue any further on its trajectory, it will reverse its direction and continue in another random direction. The ant will do this indefinitely.

When an ant finds an edge, it lays pheromone (or records the points) then it starts from its original starting point. This results in simple and fast processing algorithm. In Figure 11, the thick line represents the boundary the ant cannot pass.

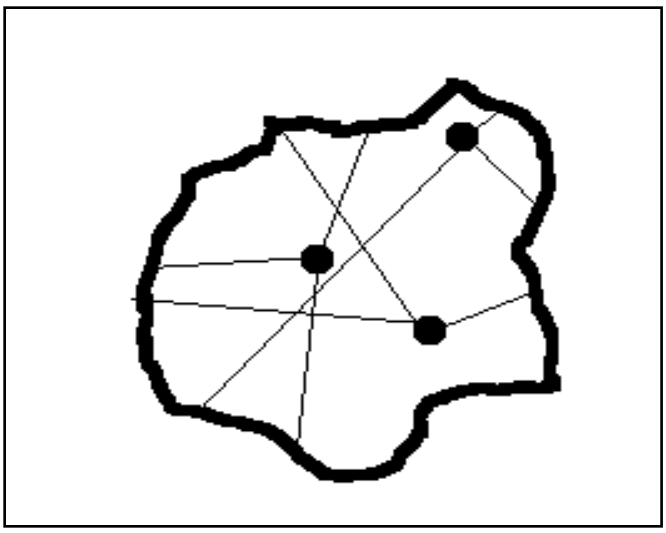


Figure 11: TAA Region

The points represent the initial random placement of the ants. Lastly, the thin lines represent a path the ant has taken to an edge. This example is contrived but develops the concept well. There are certain situations where this process will have difficulty finding a complete region which is not an issue regarding heart analysis. This difficulty is when the confined salient region is concave. Since we are working with a reasonably convex image, from the ant's vantage point, there will be very little chance of these issues. Further, the redundancy of multiple ants will eliminate most ambiguities.

TAA processed an MRI of the heart with a specified intensity (114 provided best results) and one thousand ants. The results are shown in Figure 12. The boundaries of the heart are detected well, and other artifacts are suppressed. The operation of TAA is similar to watershed in that a level was used and similar to ACO in that the image was forged to get the result.

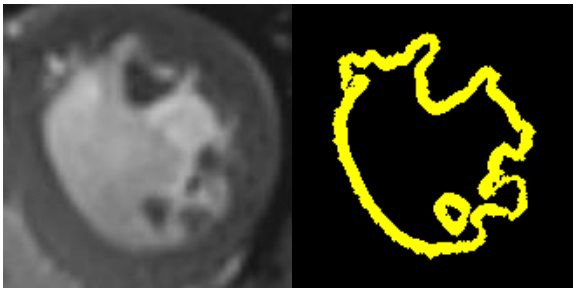


Figure 12: TAA concept

The TAA by itself is insufficient to solve any real-world problems. A main point is the threshold value was a user chosen value and not a derived value. From image to image this value needs to be updated thus needing to be intelligently derived by the algorithm. The definition of an edge can change such that the same edge may have distinctly different characteristics from one point to another. The final algorithm will derive from TAA, TACO and Stubborn ants to resolve the downfalls and provide a usable method.

### 1.2.3 Proposed Algorithm

Using concepts from earlier discussions a proposed algorithm will be introduced. It was shown that TAA is a simplistic thresholding method that does find edges but does not have a guided method to select its threshold. In TACO it was shown that thresholds can be achieved by iteration with a utility function. In stubborn ants it was shown that ants can be biased to specific paths and in elitist, some solutions can be acted on separately. The Watershed method displayed how to isolate contiguous regions called salient regions. In this research we use all discussed concepts to create a unique algorithm that

exploits salient regions, converges fast and is discriminant in the area selected. By using watershed for region selection, TACO for threshold determination and both elitist and stubborn ants for selective path generation, it is possible to find the heart regions. TAA will be the foundation of the proposed algorithm as its flexibility and extensibility allow the application of all these concepts.

To demonstrate the ease in which TAA is extensible, threshold value is automated using TACO. The image in Figure 7 was processed using this augmentation and the results shown in Figure 13.

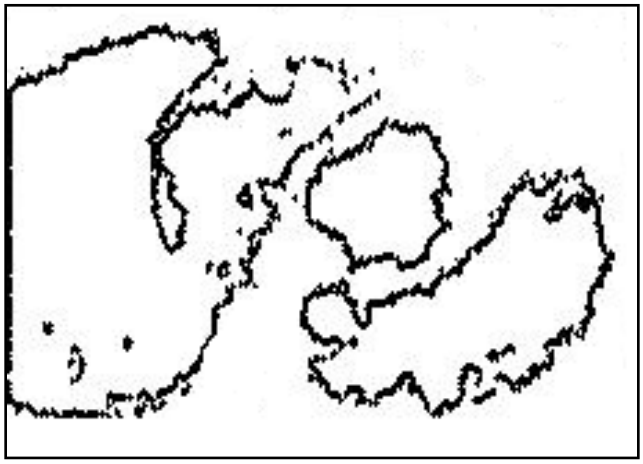


Figure 13: Reprocessed Edge Detection

The results are very good detection since predicted edges are relatively smooth and accurate. Also, unlike watershed, generation of many arbitrary cells is not an issue. Looking at the edges on a more detailed level, they do not merge with each other and the

thickness is consistently minimal. The positions of the edges are readily available to calculate volume. The right ventricle is an exemplary display of what is needed for volume calculation, but the left ventricle leaves a lot to be desired. Although this this augmentation is promising, this approach to thresholding is not robust enough to guarantee the regions are isolated.

To find the heart wall it has been discussed that a specific set of requirements needs to be enforced. It is imperative that threshold be determined by the boundary of the heart and not the characteristics of the image. Further it is important that the assumed threshold finds the exact edges with minimum deprecating artifacts. Most importantly there cannot exist segments of the heart boundary that are not found.

To resolve this, we propose an algorithm that is a merger of TAA and dynamic threshold segmentation. The previous analysis of different algorithms has shed light on what is required in a successful algorithm. In order not to be verbose, they are listed:

1. Heart edges should be differentiated from other edges
2. Edges should possess the same contours as the heart walls
3. Edges should have minimal or no discontinuities
4. The solutions should be the set of points corresponding to the heart wall

The proposed algorithm in section 2.2 conforms to every requirement in this list.

## 1.3 Evaluation and Metrics

A well know statistic will be presented using two different types of research papers working in the same problem domain. The research was selected because of their strong parallels with the current research and consistency among metrics; this is discussed subsequently. These parallels include the MRI processing medium on heart ventricles and algorithmic active contour derivation. Among both papers, as well as in image analysis, it is common to use the Dice coeffect [Dice45], which will be expanded later.

To understand the coming concepts a more fundamental concept is required which is briefly described here. When a result is said to be True positive (TP), the classification is true and, it is correct. When a result is said to be True Negative (TN) the classification is True but, it is incorrect. When a result is said to be False positive (FP) the classification is false and, it is correct. Finally, when a result is said to be False negative (FN) the classification is false but, it is incorrect.

In this section an abbreviated walkthrough of an algorithm called “Left Ventricle Segmentation in Cardiac MRI Images” [Hadhoud12] is presented as well as the identification of the metrics used.

The space is first preprocessed to develop a region of interest (ROI) in which the core algorithm operates. First, to increase consistency, only center most scans are used as this provides the best visibility of the ventricles. Using the center scans, each MRI of the sequence is processed to derive the standard deviation (SD) of intensities which is used as a metric of selection. The scan with the highest SD is selected as it is inferred that light areas (blood) and dark boundaries (tissue) are predominately distant. The MRI is segmented using the SD where 80% under the curve is assigned a class as blood and the above, classified as other.

To find the salient region each pixel has attributes extracted such as intensity, gradient, etc. The set of attributes of each pixel is large requiring substantial processing power to process. To deal with this, a scaling of the image is performed using "patches" which is synonymous with pooling in neural network (NN). Each patch has the similar attributes assigned to represent the whole set of pixels. Even with the use of patches there is still a need to further reduce dimensionality. This is done here using principle component analysis (PCA). Dimensionality reduction is a characteristic of PCA as it works by remove variance through the rotation of principle axes into the most optimal positions. The final feature representation is then clustered using K-nearest neighbors where it is assumed one cluster is blood and the other is not.

To analyze the quality of this algorithm three metrics where used called specificity, sensitivity and the Dice metric. Sensitivity is defined as TP / (TP + FN) and interpreted as

the proportion of true positives correctly identified. Specificity is defined as TN / (FP + TN) and interpreted as the proportion of true negatives identified. The dice metric was used without modifications and will be discussed below.

There are a few key differences between that research and the here proposed algorithm. In the comparative research, preprocessing is performed using a completely different technique (than the core algorithm) using a global and local minimization methods [Hadhoud11]. The proposed research here uses the same core algorithm techniques to both find the region of interest and detect its boundaries. Also, the comparative research performs dimensionality reduction by using PCA where in this research, dimensionality reduction is inherent in the linear sampling of the space.

In this section an abbreviated walkthrough of the second algorithm called "Automatic hybrid ventricular segmentation of short-axis cardiac MRI images" [Nageswararao17] is presented as well as the metrics used.

The research in this report approaches the solution by means of edge detection methods. Although there are many edge detection techniques, most work by identifying gradient changes. The method of choice is an augmentation to Kirsch edge detection with region based active contour (AC). Kirsch edge detection works by resolving a kernel using gradient analysis to project the direction of the most likely line. This happens in 45-degree rotations and thus the line will be in one of 8 directions. Region based contours is

performed by applying energy functions to the boundary. The energy functions work to become balanced by local interior and local exterior properties therefor endowing the interpolation ambiguous boundaries.

To analyze the quality of this algorithm the Dice metric and Hausdroff distance were used. Since Dice metric will be discussed subsequently only Hausdroff distance (HD) will be discussed here. At a high-level HD, is a distance function between the point of interest and the boundary it is attempting to identify. This is done by taking every point of the predicted set and finding the closest point in the reference set. Every point is normalized using the formula in Equation 7. The advantage over the Dice index is HD takes into consideration closeness of points where Dice only takes into consideration overlap.

$$MH(A,B) = \frac{1}{N_a} \sum_{a_i \in A} \min_{b_i \in B} \|a_i - b_i\| \rightarrow (12)$$

Equation 7: Huasdroff Distance

There are a few key differences between the discussed research and the proposed algorithm. First, it is assumed that another algorithm or cardiologist has drafted an initial boundary of the heart. In the here proposed research, dependency on preliminary boundary identification is not needed as well as any type of seeding. Further, the starting

boundary must be adequate to allow the energy function to converge correctly where this is not applicable here.

## 1.4 Dice Coefficient Metric

In this section the Dice Coefficient (DC) will be discussed in detail. To understand DC, it is important to know exactly what it is measuring. Let S be a set of points and let T be a set of points then $H = S \cap T$. Let S be the points of actual classification and T be the points that are of a predicted classification. This infers that S-T is the points not predicted and T-S are points incorrectly predicted. Using the non-normalized set H does not work since to the contrary, if all the points in the universe are set to true then H is S and all is predicted correctly. This means creating an over optimistic prediction set to represent the actual classification can alter the correctness of the solution. Using DC, also called Dice Metric (DM), resolves this using the formula in Equation 8 where DM $\varepsilon$ [0,1]. Where 0 is no overlap and 1 is perfect overlap. In the formula $A_a$ is defined as number of points of the area, $A_m$ is the number of points predicted and $A_{am}$ is the number of points in the intersection. This ratio implicitly has a penalization for being over optimistic or pessimistic in classification because as the denominator will become large forcing a smaller value.

$$DM = \frac{2A_{am}}{A_a + A_m}$$

Equation 8: Dice Metric

The DM provides a percentage of coverage, but it is important to statistically prove the null hypothesis, the area is detected. To do this the application of standard method in feature detection called the F test will be applied [Martin04]. Referencing Equation 9, there is a Chi test and a F test that endows this capability. In the formula the variables TPR and FPR stand for TP rate and FP rate, respectively. Further, Q is the total of true and false positive. Although this report will only contrast the two research papers with the DM, the F and X statics will be provided for thoroughness and completeness.

$$\chi^2(E_{gt}, E_c) = \frac{\text{TPR} - Q}{1 - Q} \cdot \frac{(1 - \text{FPR}) - (1 - Q)}{Q}$$

$$F_\alpha(E_{gt}, E_c) = \frac{\text{PREC} \cdot \text{TPR}}{\alpha \text{TPR} + (1 - \alpha)\text{PREC}}$$

Equation 9: Edge As feature metric

# Chapter 2
# Trapping Ant Algorithm

In this chapter the proposed algorithm will be presented called Trapping Ant Algorithm (TGAA). Take note that “trapping” is used in TGAA as opposed to “trapped” in TAA, this is to emphasizes an ongoing active process. The approach is first to rigorously extend TAA and introduce the needed argot. Then TGAA is explained and broken down into phases of operation. Each phase is introduced as a standalone process, although requires information derived in the preceding phase.

## 2.1 Extending TAA

TAA is an efficient algorithm that finds salient regions in an image and it is extended in this section to lay the foundation for TGAA. First, good parameter estimates of the boundary value and a size range of the salient region are necessary. There are five user defined parameters: threshold (H), minimum path ($t_0$), time of trail (t), number of trials (T), number of ants (N). A trial is one cycle composed of ant placement and image traversal until time runs out. The time of the trial is the maximal number of pixels an ant may travel in a trail. The minimal path is the distance an ant must travel before it is in consideration of influencing the solution. Minimal path specifies the minimum salient size to include into the result set. The threshold is the minimal intensity value the ant

considers a boundary. Please note that all values above threshold are considered a boundary. It is possible to perform TAA using other boundary specifications but is not required in this research. Ants are placed randomly at the beginning of each trial. When an ant is initial positioned, it is assigned a random direction to travel. Ants always travel in a straight line until they find the threshold. An ant may satisfy undesirable conditions which stop its processing. The following rules define undesirable conditions:

Let ant k have path $P_k$ with length $I_k$. Let ant k be randomly placed on a pixel $P_l$. Also let $t_0$ be a user defined minimal length of a path. Then there are two undesirable conditions that cause an ant to stop processing:

1. If ant k travels length $I_k < t_0$ and encounters a pixel such that H <= value ($P_l$)
2. If ant k encounters the image edge during processing time t

When an ant finds itself in an undesirable state it is said to have "died". If an ant has traveled time t without detecting the threshold, it does not provide any information and has "escaped", which is defined below:

An ant k has escaped if and only path length $L_k$ and max trail time t is $L_k = t$ and a threshold condition has not been met

Ants detect boundaries when they do not encounter an undesirable condition and have not escaped. If an ant detects the boundary and the time has not expired, it is “trapped”. An ant that is trapped is assigned a random direction and placed at its starting position to continue processing. The predicted edge solution set is defined as:

> For all trails T select all pi for all ant that do not encounter an undesirable condition or have escaped while H<value(pi) in time t. Let the set of selected pi be the solution set G

An example is shown in Figure 14 containing seven ants. The area $p \geq H$ is the boundary region where $p < H$ is the salient region. All ants initially placed in the boundary region fulfill undesirable conditions and do not travel (represented as a single dot). The ant labeled “A” has been placed in a salient region that is less than the $t_0$ and dies. The three remain ants in the salient region traverse the image adding boundary points to set G. The interpolated results of the set are shown to the right. The results are course but an accurate correlation with the salient boundary. One can see if the ants where increased to fifty that the boundaries would become more accurate.

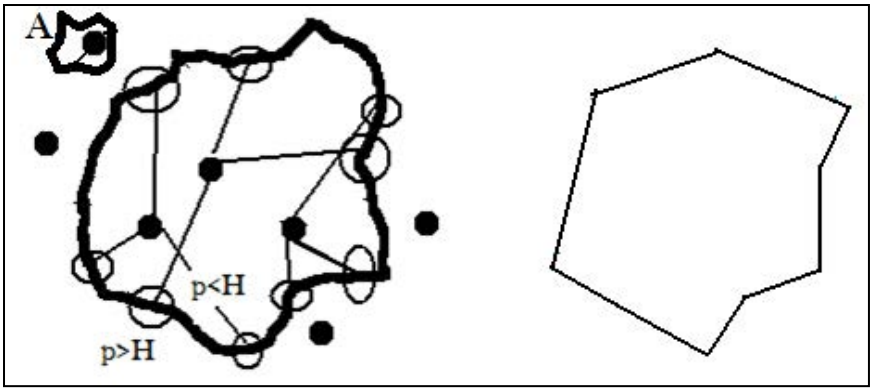



Figure 14: TAA convex hull

There are some issues still present in the extension of TAA. First a feedback mechanism is required to update the threshold. This is done using the escaped ant paths and will be performed in the Trapping phase of TGAA. Currently, TAA cannot start processing the image without user intervention. TGAA resolves initial state problem by applying a Bootstrap Phase. Finally, preprocessing akin to Gaussian, is integrated using path sensitivities. Ants do not just stop on a threshold value but require multiple thresholds, this suppress noise and other issues in multiple phases.

## 2.2 TGAA

There are four phases in TGAA. Each phase extracts information from MRI of the heart, without changing the image. A phase requires the developed information, as input, from the previous phase. In order of execution, the four phases are named Bootstrap, Trapping, Evaluation and Synthesis. Each phase will be explained in detail subsequently. The

Bootstrap phase is unique in that its only purpose is to start the algorithm and does not consume any trials or gather information that is of significant use.

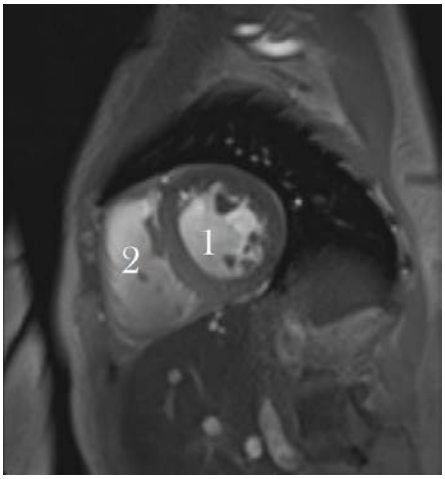



Figure 15: Reference Example

To help expedite understanding, the MRI in Figure 15 will be used throughout the algorithm explanation. The area labeled one is the left ventricle. The left ventricle has a boundary that is curvy with different grayscale values. Its salient region is consistent with some darker regions. The right ventricle, labeled two in the MRI, is similar to the left ventricle except the salient region possesses a gradual change in color. Both regions have noisy salient regions and inconsistent boundary colors that can causes serious problems when determining boundaries.

In this explanation of TGAA, we first explain ants in detail, then internal and external (user defined) variables. Finally, we visit each phase of the algorithm and display some results.

### 2.2.1 Ants

Ants are initially placed randomly, at the beginning of a trail, using a placement heuristic. Ants always travel in straight lines but the direction and initial placement change. It is important to note ants travel one pixel each time unit, this implies time and number of pixels are, somewhat, related. Therefore, when referring to ant path, time and pixel are interchangeable except an ant may have not processed the full number of pixels.

Ants require a method of boundary identification. One possible formula is distance. The pixel distance from the target salient value would remove both lower and higher pixels that are to different, refer to Equation 10.

Let B answered the question "is pixel a boundary point?" then:

$$B = \begin{Bmatrix} |S - p_c| > H & yes \\ otherwise & no \end{Bmatrix}$$

H is threshold
S is salient value
$p_c$ is current pixel value

Equation 10: Boundary

The assumption made in this research is the threshold is "darker" than the salient region. This results in the below formula:

Let B answered the question "is pixel a boundary point?" then:

$$B = \begin{cases} p_c > H & yes \\ otherwise & no \end{cases}$$

H is threshold

$p_c$ is current pixel value

Equation 11: Low Intensity Boundaries

It can be shown that several formulas may be used to manipulate detection of edges and in fact whether lighter or darker salient regions is irrelevant to functionality.

### 2.2.2 User Variables

User variables are parameters assigned before processing that influence behavior. The number of trials to perform on Trapping, Evaluation and Synthesis phases is $T_{trapping}$, $T_{evaluation}$ and $T_{synthesis}$, respectively. The total trials performed during processing are notated as $T_{total}$. As $T_{trapping}$ is increased the threshold prediction value is more precise. There is a maximum number of trails that improved threshold where any further trials will produce no benefit. As $T_{placement}$ is increased the connectivity of regions become increasingly stable representation of salient quality. There are a minimum number of

trials that must be selected to insure paths cross or the model will fail. As $T_{synthesis}$ is increase the density and continuity of the predicted edges is increased. The number of trials in phases is dependent on image size and salient qualities. The number of ants in a trial is notated by n. As the number of ants is increased processing requirements rapidly increased. It is recommended to use the least possible ants to obtain a solution although using too few ants will result in volatile results or model failure. The time of a trial is t, or maximum path length, that an ant can travel. The trail time must be greater than the salient region but smaller than the image size. The maximum salient size to consider is $t_{max}$ and is valid in the placement phase only. The quality of a salient region is quantified by an integer number and called connectivity. The user defines the number of best-connected salient regions ($C_{sel}$) that should be used in the solution. The target region is a numerical value representing intensity notated by $I_{target}$. The exact value is inconsequential as the algorithm is robust, but it is required that the value be less that the boundary value. The exploratory and exploitation characteristics of the ant can be modified by $\tau$ such that $\tau > 0$. Ants may encounter noise in the salient region and falsely mark pixels as an edge. To reduce these ants can be assigned threshold sensitivity of $t_{sense}$. The use of $\tau$ and $t_{sense}$ will be explained in the placement phase section.

### 2.2.3 Internal Variables

The following variables are created and updated, by the model, during the run. Each ant is assigned a direction (v) in which it travels at the beginning of each trial. This random

variable is in degrees implying $0 \leq v < 360$ and does not change during a trial. At the end of a trail ants are in one of three possible states. An ant is either dead, escaped or trapped where the number of ants in these states is $n_{dead}$, $n_{escaped}$ and $n_{trapped}$, respectively. Elaboration of states will be explained further in the trapping section. The current value of the threshold is notated by H and is monotonically changing. Let $L_{min}$ and $L_{max}$ be the minimal and maximal value of a pixel in an image then $L_{min} \leq H \leq L_{max}$. The connectivity of a specific salient region is notated by $C_i$ and the number of total predicted salient regions is $S_{total}$. The placement phase records all ant paths that are trapped this set is notated as $P_{trap.}$ At the end of this phase redundant and pixels greater than the threshold are removed from $P_{trap}$, this set is notated as $P_{prune}$. Therefore, in normal circumstances $| P_{trap} | >> | P_{prune}|$ but it is always the case that $| P_{trap} | \geq | P_{prune}|$. The evaluation phase records all paths that have crossed and is notated by the set $P_{place}$. The selected set of paths meeting minimum connectivity criterion is $P_{sal}$ and identifies all possible salient prediction sets. Each salient region has $C_i$, the number of path crossing, associated with that region. Therefore, in normal circumstances $| P_{sal} | >> | P_{place}|$ but it is always the case that $|P_{sal}| \geq |P_{place}|$. The evaluation phase records all endpoints of paths and is notated by set $P_{eval}$. The set conditioned to remove all redundant nodes resulting in the solution set $P_{final}$.

### 2.2.4 Bootstrap

In the Bootstrap phase an initial threshold value is generated, $H_{start}$ ,then assigned to the global threshold H. TGAA algorithm cannot begin execution without a starting threshold. An H value is required because it determines if an ant has stopped by encountering a boundary condition. Assigning $H_{start}$ to the maximum possible intensity allows a state in which all ants can escape. The update heuristic is unrestricted causing the initial H value to possibly be to low intensity. This is a result of the algorithm working by monotonically decreasing the threshold H therefore it is imperative that threshold be selected higher than the salient region and lower than the maximum intensity in the image. The starting threshold characteristics is shown in Equation 12.

Let $H_{start}$ be the starting threshold, $H_{salient}$ be the salient threshold.

$$H_{Salient} << H_{start} < \max(\forall \text{ pixels})$$

Equation 12: Starting Threshold

Therefore, the algorithm requires a starting point that is not part of the standard algorithm. The resolution is a simple bootstrap technique. Before trial one the image is traveled by ants in random directions with no applicable heuristics. The only rule is ants

that touch edges are eliminated from this sample. This eliminates the possibility that an ant may be spawned next to the edge reducing the sample size causing possible biases. The highest pixel value of all paths is assigned as the boundary. This value will be significantly higher than the boundary but represents an arbitrary starting point. The algorithm is summarized in figure 16:

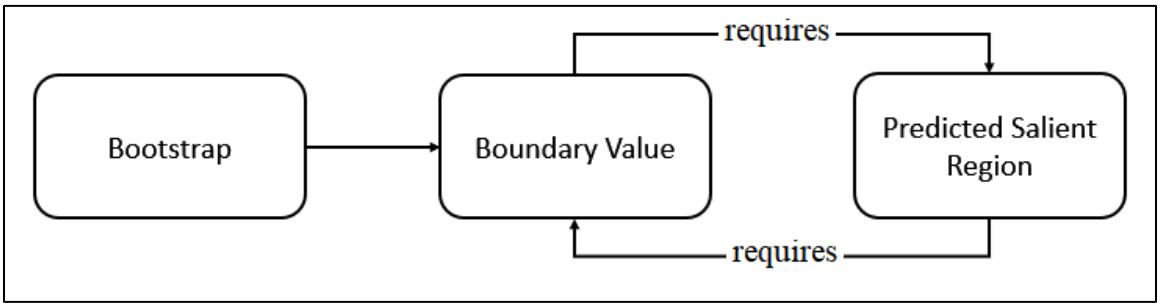



Figure 16 Bootstrap

### 2.2.5 Trapping

The trapping phase performs two activities. First, information gathering and the second, data cleaning (pruning). Together, these activities survey the image while, simultaneously, conditioning preliminary solutions. After each trail paths are dichotomized into the sets trapped ($P_{trap}$), escaped ($P_{escaped}$) and dead ($P_{dead}$), see section 2.1 for more information. Subsequently, it will be shown how $P_{trap}$ and $P_{escaped}$ are formed and used. In regard to dead ants, they are not processed and disregarded. Before discussing these sets, however, sensitivity needs to be discussed.

Due to image noise, stopping immediately on a boundary condition is not fruitful and requires detection to be somewhat impervious to these conditions. Noise issues were discussed previously using Gaussian masks; here the image is not preprocessed but performed dynamically during model execution. Noise is dealt with using ant sensitivity, as defined in this research in Equation 13.

Constraint: Path $P_k$ is an element of $P_{trap}$ if and only if there exists $p_i \varepsilon P_k$ such that:

$P_i, P_{i+1} \ldots, P_{sense} \geq H$

Equation 13: Trapped Path Threshold

The output of sensitivity is a simple binary value and input is pixels for a duration in the past, starting with the current time. If for any sequence of pixels of size n with a pixel value below threshold the function returns false. Referring to Figure 17, letting s = 5 (sensitivity) and each line of boxes be a path traveling in direction of the arrow. The grey boxes represent above threshold and the white boxes below. The number inside the boxes represent the consecutive pixels greater than the threshold. In the top path, the sensitivity is incremented in the second box to one but resets in the third because the threshold was not met. The sensitivity has a largest value of 3 which is less than 5, implying thresholds are not consistent enough. In the bottom path, boxes 4 thru 8 are all sequentially greater than the threshold resulting in the sensitivity of 5, meeting the minimum sensitivity constraint. This results in threshold detection and the function returns true. A subtle point to clarify, the actual boundary starts at the beginning of the sequence detecting the edge. The “start” in the bottom path represents the boundary.

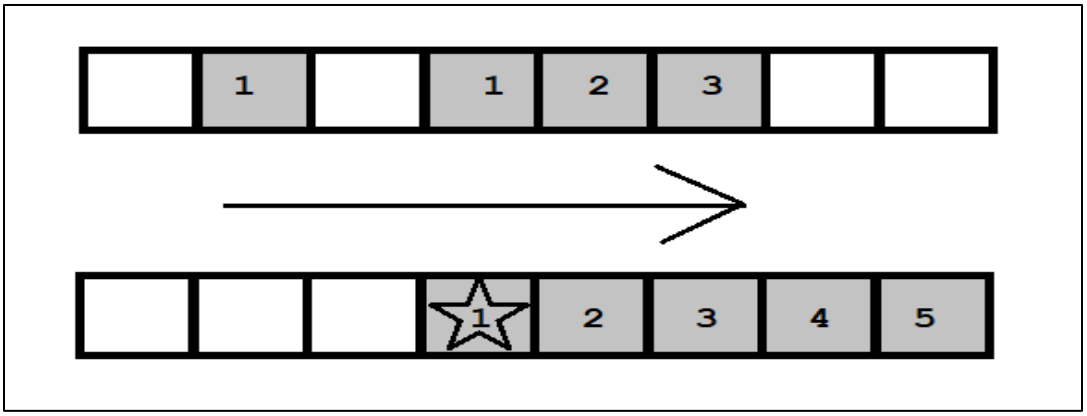



Figure 17: Threshold Sensitivity

Information gathering begins with an empty salient prediction $P_{trap}$, Bootstrap threshold (H) and the user defined number of trials ($T_{trap}$). The salient prediction region is a set of paths aggregated in $T_{trap}$ trials. The paths are added to the set $P_{trap}$ as defined in Equation 14. For further foundational information refer to Equation 8. Alternatively, a path is added to the trapped set if it did not die or escape.

For trial T in $0 < T < T_{trap}$ let k be the $k^{th}$ ant with path $P_k$ and path length $L_k$. Let $P_k$ be qualified such that $L_{min} < L_k < t_{max}$ and $P_k$ did not encounter an image edge.

Constraint: Path $P_k$ is an element of $P_{trap}$ if and only if boundary sensitive is met

Equation 14: Trapped Paths

At the beginning of this phase the escaped set $P_{escaped}$ is empty. An ant path is added to $P_{escaped}$ if the escaped path, shown in Figure 18, holds. Alternatively, a path is added to the escaped set if it did not die and it was trapped.

For trial T in $0 < T < T_{trap}$ let k be the $k^{th}$ ant with path $P_k$ and path length $L_k$. Let $P_k$ be qualified such that $L_k = t_{max}$

Constraint: Path $P_k$ is an element of $P_{escaped}$ if and only if boundary sensitive not is met

Figure 18: Escaped Paths

At the beginning of this phase ants are placed to foster exploration. Exploration is implemented by randomly placing ants throughout the whole image at the beginning of each trial. The exploratory behavior allows a survey of the image while reducing the possibility of falling into a local minimum. Exploitation, by use of Stigmergy, is the result of placing ants randomly in $P_{rap}$ at the beginning of each trail. The proportion of ants randomly placed anywhere in the image versus placed randomly in $P_{trap}$, is a heuristic defined in Equation 15.

$$Pr_{init,sal} = \begin{cases} \eta(T_{current}) & if\ 0 \leq \eta(T_{current}) \leq 1 \\ 1 & if\ \eta(T_{current}) > 1 \\ 0 & otherwise \end{cases}$$

$$\eta(T_{current}) = \frac{T_{current} - \tau}{T_{trap}}$$

Where

- $Pr_{Int,sal}$ is the probability a ant is initially placed in the salient prediction set $P_{trap}$.
- $T_{current}$ is the current trial
- $T_{trap}$ is total number of trapping phase trials
- $\tau$ is offset where $\tau > 0$ and an integer
- $\eta$ placement function

Equation 15: Placement Heuristic

For trial one, η is zero implying $Pr_{init,sal}$ is zero forcing exploratory behavior not to occur. This must be the case since the first trial has an empty salient prediction set. As τ increases the behavior is more exploratory increasing both the number ants randomly placed and trails. Conversely, as τ decreases the behavior is more exploitive in both number of ants and trials. The two behaviors have an inverse relationship expressed through τ, but one can increase both behaviors through the increase of number of ants per trial.

As trials increase, the size of sets $P_{trap}$ and $P_{escape}$ increases where at the beginning $P_{escape}$ increases rapidly and towards the end $P_{trap}$ does. This is pertinent as information is leveraged in a similar fashion such that at the beginning trapping escaped ants is critical to threshold development and toward the end discovering the boundaries using trapped

ants is priority. The threshold is updated after each trial using $P_{escape}$ and is monotonically decreasing. The update is performed using Equation 16.

Threshold is updated as follows: $H = \min (\forall p_{k,max} \varepsilon P_{escape} \cup H)$

Equation 16: Threshold Update

This simply states the minimum of all maximum values may update the threshold only if it is less than the threshold. The new threshold causes every ant that traversed these paths to become trapped. To support solidarity and understanding refer to Proof 1.

For some trial let P be the set of all path escaped and $p_i$ be the $i^{th}$ path. Let $\grave{p}_i$ be the max intensity of path i.

$$H = \min(\forall \grave{p}_i)$$

Then it must be the case for all paths in p $\exists$ $p_i < H$ and thus every ant is trapped.

Proof 1: Trapping Escaped Ants

Finally, as a corollary of Proof 1 and Equation 16, every ant in previous trials remains trapped, refer to Corollary 1.

In Equation 16 threshold H is shown to be monotonically decreasing for every trial and in Proof 1 every ant is trapped when H is updated.

Therefore $H_{t+1} \leq H_t$ implying $\min(\max(p_{t+1})) \leq \min(\max(p_t))$ and therefor all previously escaped ants are trapped.

Corollary 1 Trapped Ants Remain Trapped

The results of the trapping phase, the set $P_{trap}$, is shown in Figure 19.

As shown the paths have converged to the lighter regions where they cluster or become denser. Further inspection displays that salient regions have been identified for multiple intensity levels that are greater than the heart. Although the heart is strongly detected the paths protrude out of the salient region into the boundary. To resolve these unwanted attributes cleaning is performed next.

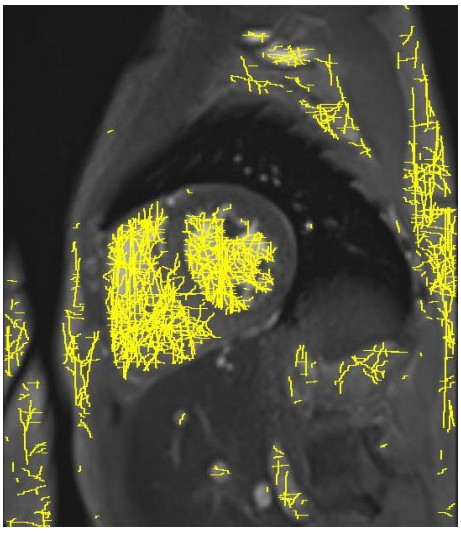


Figure 19: Path Convergence

Cleaning $P_{trap}$ is necessary since the current solution does not adequality reflect the final threshold under normal model operation. To understand this an additional specification is outlined in Corollary 2. The implication of this corollary trapped is that ants may have pixels above the current threshold and are no longer valid in the final solution.

| |
|---|
| As prescribed by Proof 1, let trial $T_n$ have escaped ants trapped with threshold $H_n$ and trial $T_{n-1}$ have escaped ants trapped with threshold $H_{n-1}$<br><br>If $H_{n-1} < H_n$ (Equation 16) then it is possible for paths in $T_{n-1}$ to have higher intensities than $H_n$ |

Corollary 2: Trapped Ants above Threshold

In the cleaning activity, named pruning, the result set $P_{trap}$ is reprocessed where no trials are performed or consumed. There are two issues to resolve, the previously described path threshold problem and duplicated pixels (same location) from paths crossing. In the first, the reprocessing of all paths with the new threshold is performed. In the second, removal of redundant pixels eliminates bias during random placement of ants. The result is the qualified prediction set $P_{prune}$ and contains all usable salient prediction paths. The pruning of the predicted salient set is shown in Figure 20 .

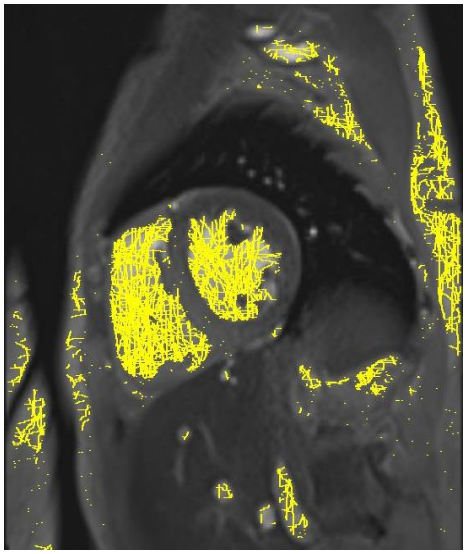


Figure 20: Path Threshold

It is shown that the paths between salient regions have been removed. Further artifacts that held no value have been effectively removed. The MRI displays salient disjointedness and convergence on possible salient regions. Given this information, it is now possible to quantify the quality of the salient regions.

### 2.2.6 Evaluation

The evaluation phase consists of two different activities. These activities are data generation and evaluation. Generation of data, salient region paths, is done by performing TAA a user defined $T_{place}$ trials. Evaluation of the paths provides new information to make quantifications about the salient region.

At this point in the algorithm the salient regions have been found and a threshold established. In the data generation activity ants perform a TAA where initial placement is randomly assigned in the set $P_{prune}$ and the threshold value is H. This activity requires enough trials to ensure that paths cross in the salient regions. As trials increase, the number of path crossings increase, providing a more stable solution. The paths of every trial are aggregated into the results set $P_{place}$.

In the evaluation activity the connectivity of all prospect salient regions is calculated. This is done by comparing all paths in $P_{place}$. If two paths cross, they are in the same salient region and the connectivity of the region is increased by one. Connectivity is defined here as: crossing of paths generated from $P_{place}$, either directly or indirectly, limited by H. By "indirectly crossing" it is meant that transitivity applies. The individual predicted salient regions are quantitatively separated and have an associated connectivity. A new set $B_T$ is introduced in Equation 17 to facilitate understanding:

| Let $B_T$ be a set of order pairs (B, C) where B is a set of paths and C is the connectivity of those paths |
|---|

Equation 17: Connectivity

Every single predicted salient region B is disjointed from others. This is by definition and is explained in Equation 18:

Let $P_j$ and $P_k$ be any paths that updated $B_T$ where $B_T = \{(B_1,C_1), (B_2,C_2),\ldots(B_n,C_n)\}$ then the following holds: if $(P_j \cap P_k) \neq 0$ Then $\exists(B_r, C_r) \in B_T \; S.T. \; P_i \cup P_k \subseteq B_r$ and $B_r \cup_{p \neq r} B_p = \{\}$

Equation 18: Disjointed Regions

It should be noted that a prediction set B may have the quality that there exists a salient region such that more than one B belongs. If this is the case the more connected prediction region is given precedence. The connectedness of a region is the number of crossing in a salient prediction set. This is defined Equation 19:

For every $(P_j \cap P_k) \neq 0$ in $B_r$, $C_r$ is incremented by one.

Equation: 19: Connected

In the ongoing MRI example, the number of most connected regions to consider ($S_{color}$) is two. As shown in Figure 21, there are eight regions selected. These are the most highly connected salient regions. As shown every path is connected to another.

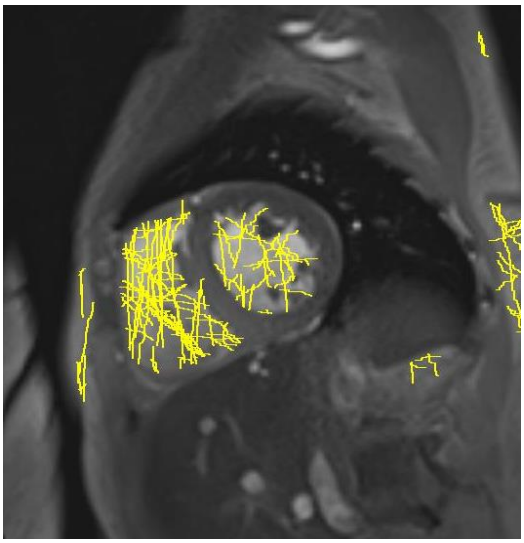


Figure 21: Evaluation

### 2.2.7 Synthesis

The synthesis phase builds the final solution by finding salient edges. This phase performs a user defined $T_{syth}$ trials. In the evaluation phase, $C_{sel}$ number of predicted salient regions were chosen and the respected paths aggregated to $P_{eval}$. This phase performs a TAA with ants randomly placement in $P_{eval}$ with threshold set to H. The threshold location found by each ant after a trial is aggregated into $P_{eval}$. This produces a growing set that will detect edges normally unfound. At the end of this phase all redundant points are purged resulting in the solution $P_{final}$, as demonstrated in Figure 22.

Figure 22: Synthesis

### 2.2.8 Visual Representation

To help visualize the flow and interconnections of TGAA, refer to Figure 23 below

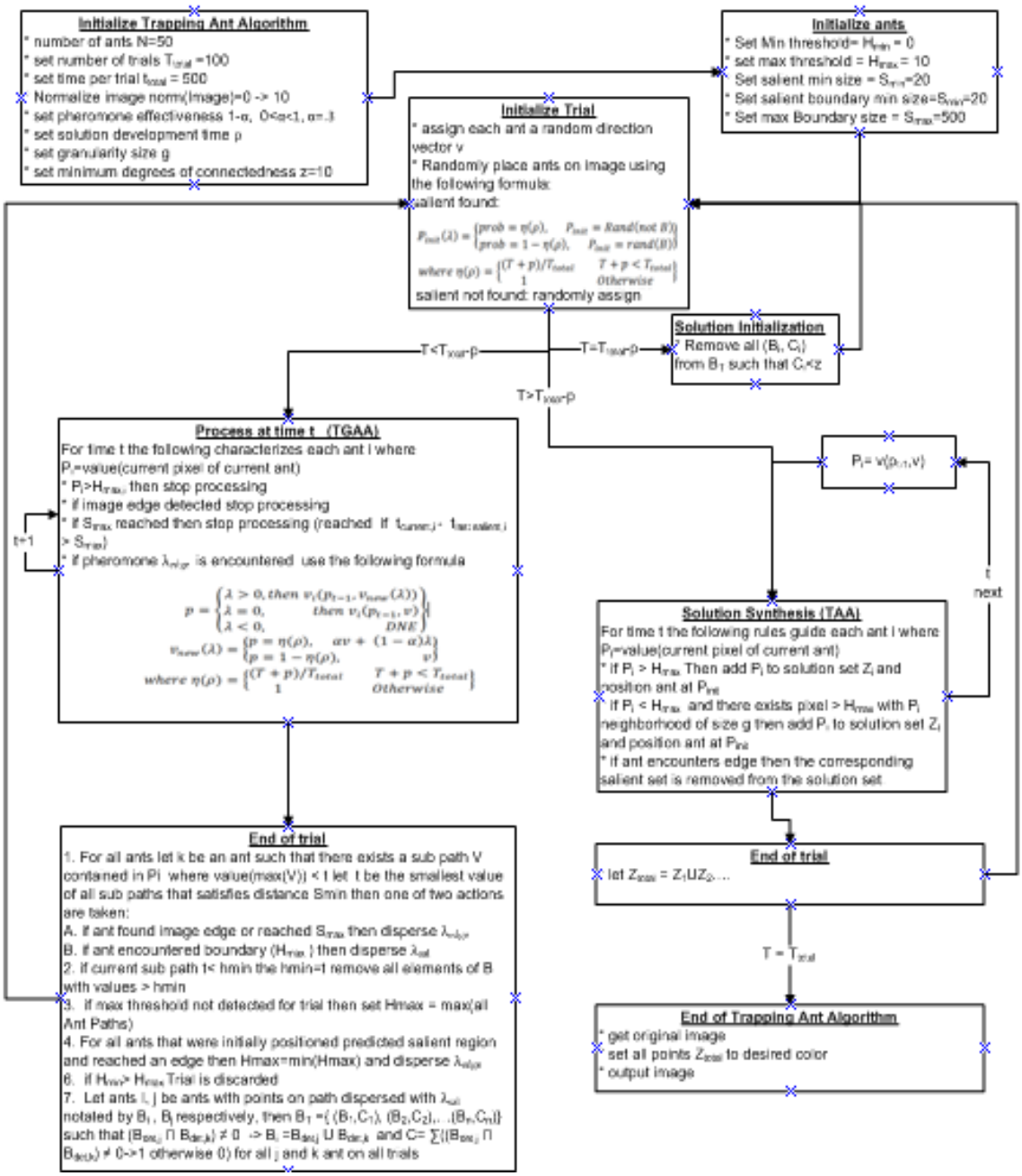



Figure 23 TGAA Visual Representation

# Chapter 3

# TGAA Analysis:

In this chapter analysis will be conducted to assess the efficacy of TGAA. First, a discussion of how the sample data was acquired, cleaned and disseminated. Then analysis is performed on the sample data for quality and predictions correctness. Finally, a conclusion is made as to the capability of TGAA.

## 3.1 Sample Data

In this section a discussion of the procurement and conditioning of MRI image samples is given. The procurement of MRI images has to be done in a way that represents the population and assigned so that no biases occur. Special attention has been given to the procurement and is described in detail below. To perform testing the images have to be transformed, or conditioned, into a format that can be compared against the prediction set using the DM. Care has been given not to alter the sample results and the transformation will be explained as well.

As describe previously the full set of MRI images contains different slices of the heart

across time. Also, as previously discussed in the refence report “Left Ventricle Segmentation in Cardiac MRI Images”, some slices of the heart do not provide information or are too noisy to process [Hadhoud12]. These issues are normally on the boundary slices; in this research the boundary slices are 1 and 7. It was found, by visual and preliminary processing, that slices 6 and 7 were not able to be processed although slice 1 had enough information. Further, in most cases these slices of the heart did not contain blood. Reviewing Figure 24, one can see there is no light regions, this implies absence of blood only regions which has no volume.

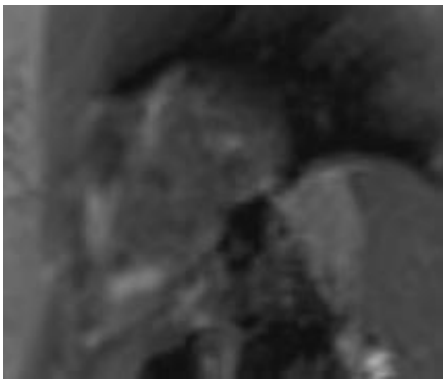


Figure 24 Slice Six

The sampling is done using 12 subjects to manually draw, what they perceive to be, the walls of the heart. The subjects where not coached on how to identify the heart but only on the drawing tool use and what an MRI entails. To evaluate the algorithm, a fully randomized sample of 48 MRI images was procured from a full set of 850. These 48 MRI images where then randomly assigned to 12 subjects in which every image belongs to 3 different subjects. This results in a validation set of 144 MRI images where each subject

has 12 MRI images to work on. For more information of setup and design refer to the Sample Assignments section in Appendix A.

After the images were collected, conditioning was performed to provide a consistent format. This was done by extracting all yellow information from MRI images samples. All non-yellow pixel was set to white and the inner contiguous regions were set to yellow. This is shown in Figure 25 where the processing happens from left to right

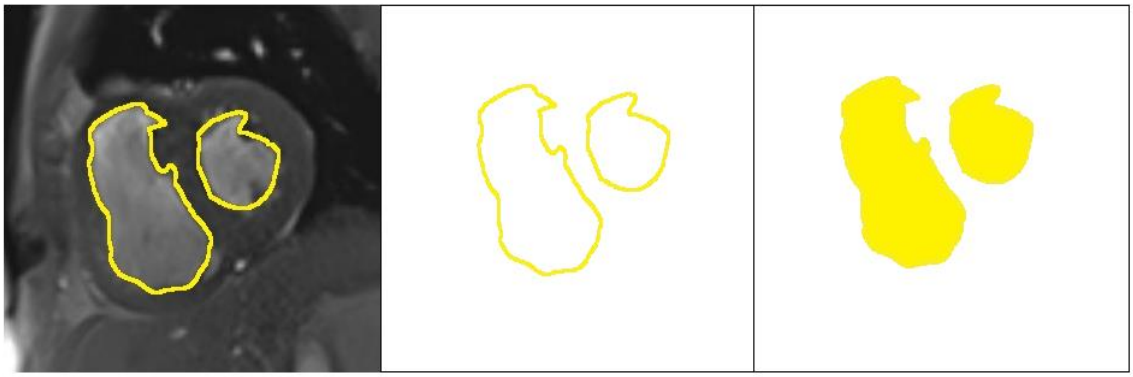


Figure 25 Image Processing

## 3.2 Results

In this section an analysis of the algorithm contour detection capability is performed. As discussed in the metrics section assessment will be performed using the Dice Metric (DM). To do this the samples are first evaluated as to consistency and viability. This evaluation is done at both the human subject level and MRI images. It is possible for the subject to not grasp or mechanically perform accurate boundary detection. Also, some MRI images may be confusing for all subjects causing low consensus and little

agreement. Last, several experiments were performed to understand the contour identification capability of the algorithm.

To understand the constancy of the subjects and the resulting MRI samples, analysis will be performed to identify any issues. It is important that there is consensus by the subject's results image wise. A comparison is done for each MRI across all the subjects' samples and results are shown in Figure 26. A horizontal line is drawn to highlight the DM below a reference value for outliers using the standard formula max(min(x), Q_1 – 1.5 * IQR)). The value Q_1 and IQR were derived from all MRI samples resulting the outlier value of 80.8%.

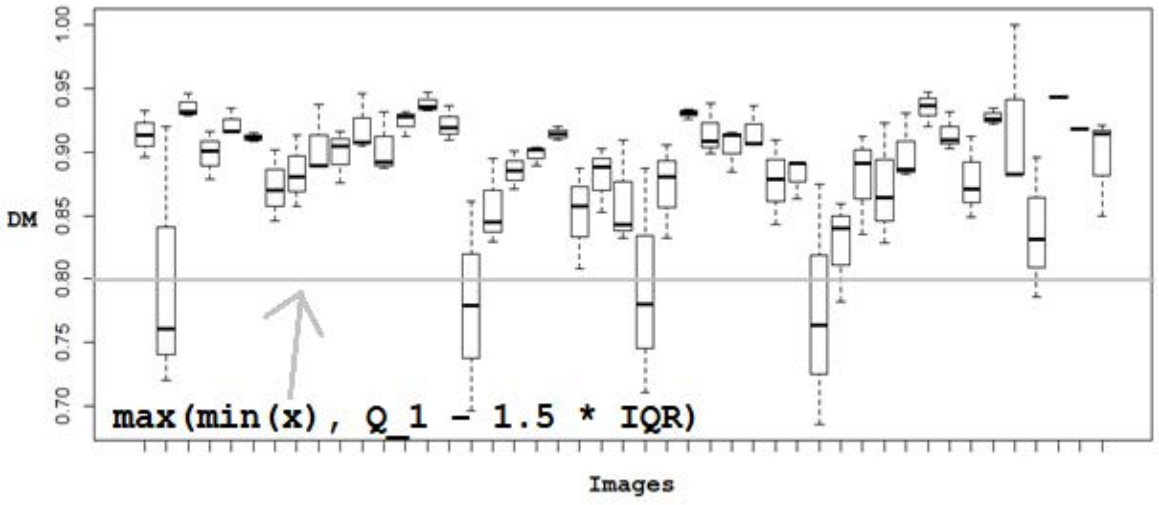



Figure 26 DM by Sample by Image

It is shown that for six MRI groups DM was lower than 80.8% and totaling10 subject samples. Also there seems to be a large variance across images, some boxplots have a

very large range, and some have very narrow. As the boxplot becomes narrower it can be said the consensus of the MRI is high. Alternatively, if the boxplot is wide, the subjects have less consensus. The largest disagreements happens to also be in images that cross the anomality boundary. Given this, analysis of intra image variance will be refrained from.

**DM < 0.8**

| # | DM | Subj. 1 | Subj. 2 | image |
|---|---|---|---|---|
| 1 | 0.761 | 6 | 4 | 24 |
| 2 | 0.721 | 6 | 5 | 24 |
| 3 | 0.780 | 8 | 9 | 13 |
| 4 | 0.710 | 7 | 8 | 13 |
| 5 | 0.779 | 7 | 6 | 21 |
| 6 | 0.70 | 8 | 6 | 21 |
| 7 | 0.790 | 6 | 4 | 22 |
| 8 | 0.764 | 6 | 7 | 38 |
| 9 | 0.686 | 8 | 6 | 36 |
| 10 | 0.782 | 6 | 4 | 14 |

**Discrepancies User**

| User # | Count |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 3 |
| 5 | 1 |
| 6 | 8 |
| 7 | 3 |
| 8 | 4 |
| 9 | 1 |
| 10 | 0 |
| 11 | 0 |
| 12 | 0 |

**Discrepancies Image**

| Image # | Count |
|---|---|
| 24 | 2 |
| 13 | 2 |
| 21 | 2 |
| 38 | 1 |
| 36 | 1 |
| 14 | 1 |
| else | 0 |

Table 1: DM Outliers for MRI samples

It is shown in the Table 1 that Subjects 6 & 8 contribute to 60% of the outliers implying there is a possible significant difference in their samples. Since the distribution is unknown, or at least not well defined, a non-parametric Chi Squared test is performed. The only assumption made is that all subjects have the same probability of having outliers. Referring to Table 2 test number 1, it is shown that the null hypothesis has been rejected in favor there exists a difference in proportion of outliers by user. Replacing the

discrepancies with the expected median value outlier (0.5) for subjects 6 and 8 it is found in test 3 the test fails to reject the null hypothesis that discrepancies of subjects are significantly different.

| Test # | Sample Description | X-squared | DF | p-value |
|---|---|---|---|---|
| 1 | All | 40 | 11 | <.001 |
| 2 | Student 6 median | 21 | 10 | 0.021 |
| 3 | Subject 8 & 6 median | 18.33 | 11 | 0.074 |

Table 2: Subject Chi Squared test

Due to the limited number of samples, removing two subjects, or 24 MRI images, is a significant obstruction. Therefor a compromise is made to remove subject 8 and continuing with subject 6. Performing Chi Squared with subject 6 in test 2 results in a p-value 0f 0.021.

Every algorithm identification compared against the sample resulting in 136 DM values. Once again, A horizontal line is drawn to highlight that all DM below this value are outliers using the standard formula max (min(x), Q_1 – 1.5 * IQR)). The value Q_1 and IQR were derived from all MRI predictions resulting the outlier value of 68.1%. The results are shown in Figure 26 where it is readily apparent there are 3 outliers.

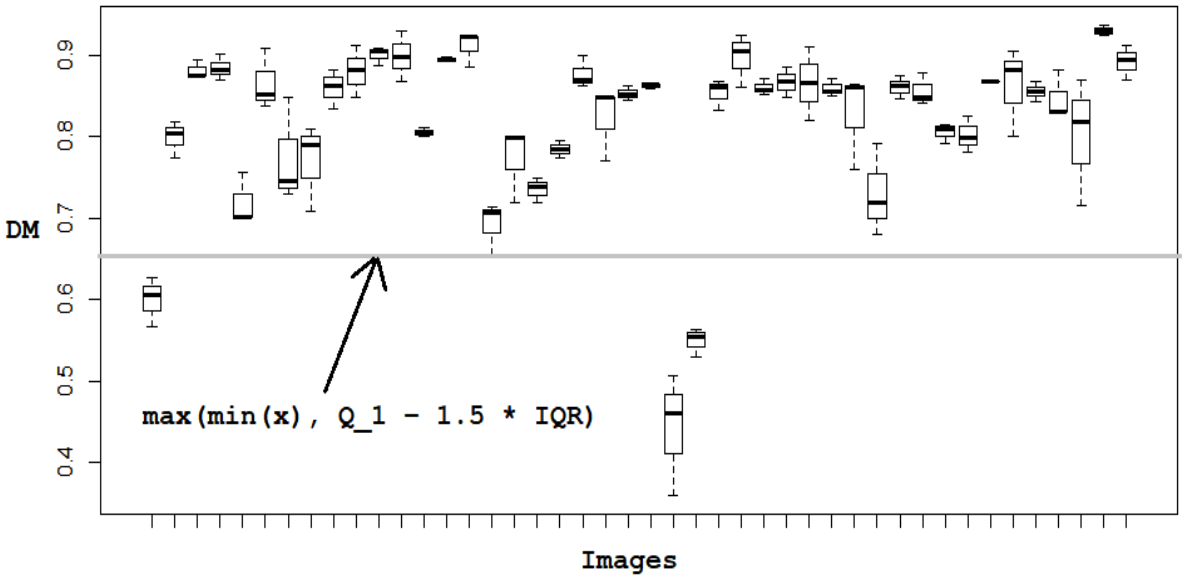



Figure 26: DM by Predicted compared to samples

Focusing on the outliers the results are tabulated in Table 3 where it is shown that three images cause 100% of the issues. Subjects 1 and 12 have two samples as outliers but 2 outliers in this context turns out not to be significant and were not pursued. Also, it is interesting that for these images all three samples were different. This means, there is a consistent disagreement among the prediction ant sample set.

| Compare # | DM | Subject # | Image # |
|---|---|---|---|
| 1 | 0.606 | 1 | 39 |
| 2 | 0.567 | 12 | 39 |
| 3 | 0.627 | 11 | 39 |
| 4 | 0.461 | 9 | 13 |
| 5 | 0.361 | 8 | 13 |
| 6 | 0.506 | 7 | 13 |
| 7 | 0.530 | 12 | 23 |
| 8 | 0.564 | 1 | 23 |
| 9 | 0.554 | 2 | 23 |

| image # | issue count |
|---|---|
| 39 | 3 |
| 13 | 3 |
| 23 | 3 |
| All else | 0 |

Table 3: Prediction Outliers

Example outlier (MRI #23) is selected as a paragon because it displays two issues that are present in all other outliers. Referring to the raw classification (full paths shown) in Figure 27, there are two regions identified that subjects did not identify. The region labeled “Issue One” has the same intensity of blood and same geometric properties. Issue one could have been resolved by reducing the region of interest. The region labeled “Issue Two” is correct as the heart is malformed, and the subjects were unable to discern this.

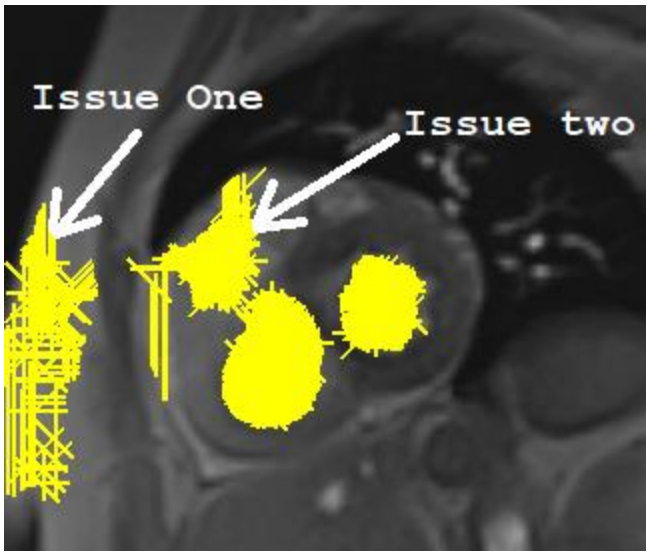



Figure 27: MRI outlier Example

In order to provide complete transparency, analysis was performed on all the data with outliers not remove, called “Raw” and with outliers removed Called “Conditioned”. The descriptive results are shown in Table 4 where one can see there are differences in means in both raw and conditioned data. Further, variance seems to be larger for predictions in both cases.

| | Raw | | Conditioned | |
|---|---|---|---|---|
| | Samples | Prediction | Samples | Prediction |
| mean | 0.888 | 0.817 | 0.902 | 0.844 |
| 1st Q | 0.8709 | 0.791 | 0.884 | 0.810 |
| Variance | 0.002748543 | 0.010 | 0.001 | 0.003 |

Table 4: Summary of DM by Set

Tests were performed to evaluate the differences in means between the samples and predictions for DM. This was done using a more sensitive T test that assumes some

normality in the underlying distribution and a Wilcox Ranked test with no assumptions of the underling distribution. Further, these tests were done on the all data and conditioned data resulting in four tests.

Referring to Table 5, it is shown that both tests reject the null hypothesis in both conditioned and raw data sets. Therefore, we can conclude that prediction set is significantly different from the subject sample set.

| | T Test | | Wilcox Test | |
|---|---|---|---|---|
| Description | Df | p-value | W | p-value |
| All Data | 198 | <.001 | 3780 | <.001 |
| Outliers Removed | 179 | <.001 | 2064 | <.001 |

Table 5: T test difference of means

Although it has been shown that the prediction set does not match the MRI samples the question remains: Does this algorithm provide any prediction capability?

To answer this question a bootstrap experiment was performed. This was done by comparing DM of random subject samples against other subject samples, where the samples were not the same. This results in a DM that reflect a random prediction on a subject sample. If the DM for a random prediction on a sample is significantly different

from the predicted DM, there would still be benefit in the algorithm. Further, if there is a significant difference, it would follow that the algorithm provides more power than random predictions but less than human predictions.

To do these, 30 subject samples were selected randomly and then compared against each other. The results are summarized in Table 6, which shows non-zero but very low mean. Further the variance is significantly higher than all the previously discussed summaries.

| | |
|---|---|
| mean | 0.263 |
| 1st Q | 0.06 |
| Variance | 0.064 |

Table 6: Summary Random Sample Comparisons of DM

To visually understand the difference of the results, refer to Figure 28. One can see the "boot" (representing the bootstraps) set has much greater variance and lower mean than both prediction and sample sets.

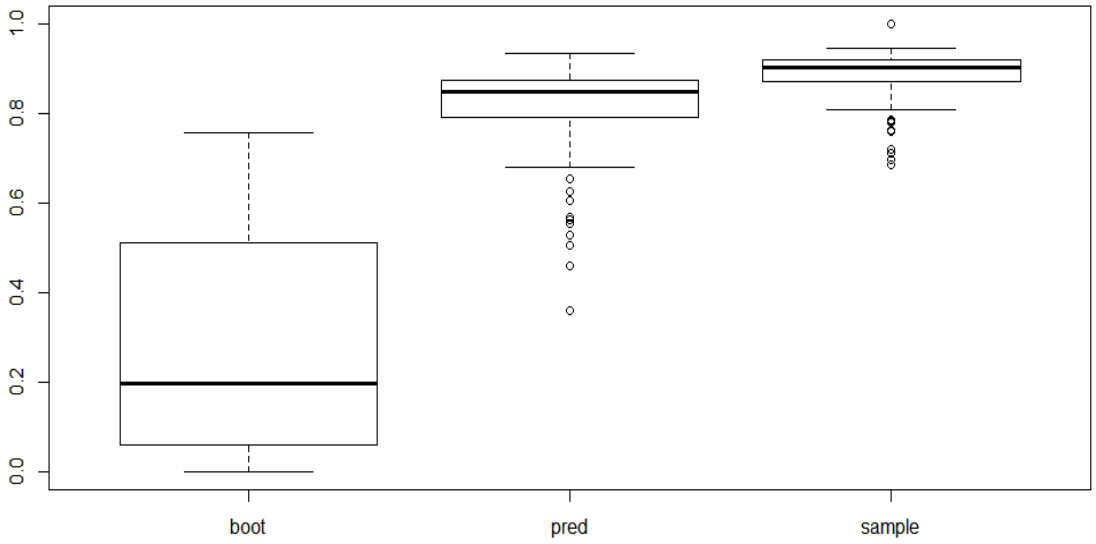



Figure 28: DM by set type

As before, the bootstrap set of DM will be tested used the T test and Wilcox Rank test. Table 7 displays the results of the tests. Both tests fail to accept the null hypothesis of similar means in favor of the alternative that means are different. Therefore, it can be concluded the algorithm does predict heart volume better than random predictions of the heart volume.

| | T Test | | Wilcox Test | |
|---|---|---|---|---|
| Description | Df | p-value | W | <.001 |
| All Data | 24 | <.001 | 75 | <.001 |

Table 7: Test Random Samples vs Prediction

## 3.3 Conclusion

TGAA does have the ability to find regions in the heart. This was shown using the dice metrics to compare the prediction set against inner random samples comparisons. Although shown to have prediction power, TGAA was not able to predict at the same accuracy as human subjects. However, there was 6% difference in the DM which equates to 6% difference in volume and may prove to be of minimal impact in real world settings. Also, TGAA worked on a variety of sample MRI in both size, slice angles and heart topology differences. Therefore, it is concluded that TGAA has prediction power but not as strong as a human subject.

Chapter 4

Future Research

In this research analysis was done only on the path. It is possible to perform analysis on a sliding window (neighboring pixels) as ant travels. Performing a proof of concept, we found a breast tumor shown below in Figure 29.

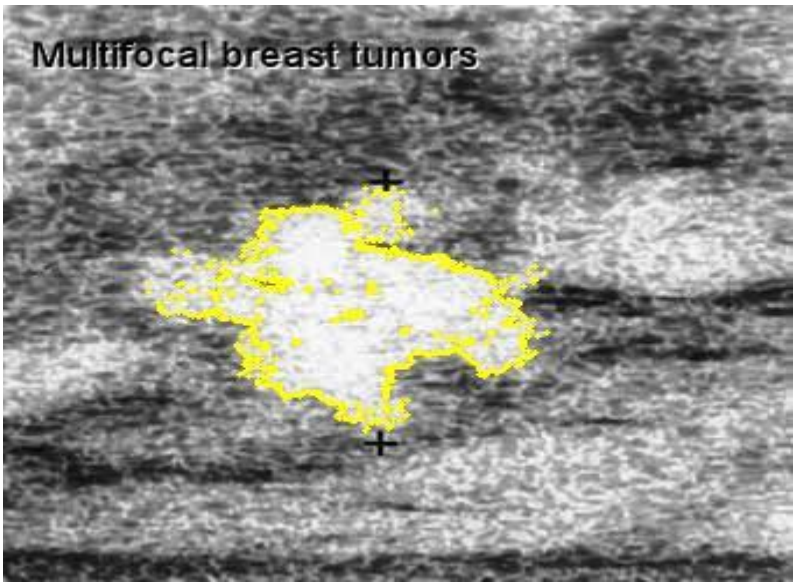



Figure 29: Tumor POC

TGAA was augmented with a window of size 20X20 and a density function as trapping condition. Also, there may be useful information in the natural clustering that occurs when this algorithm is run as shown below in Figure 30.

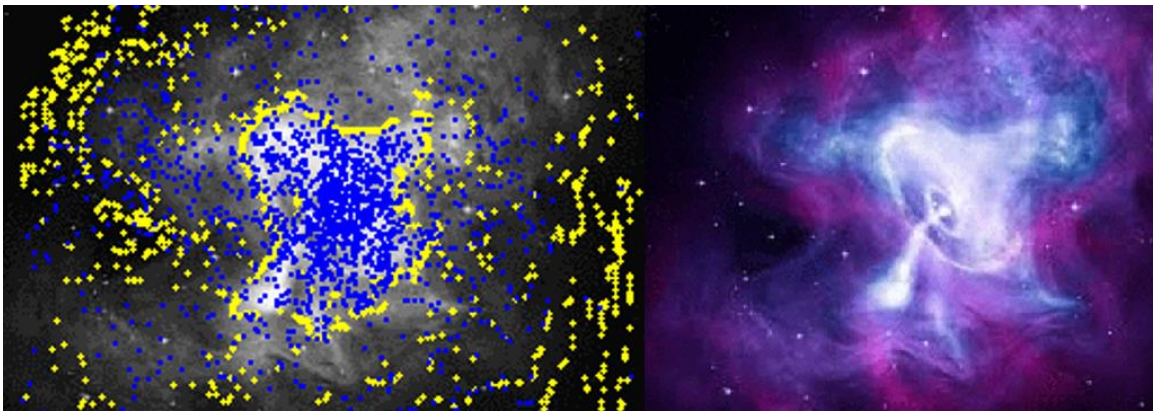


Figure 30: Clustering POC

The yellow points are end points of all trapped paths and the blue are starting points. We see lines are developing within the yellow points and salient regions within the blue.

Lastly, TGAA lends itself well to parallel processing. Trials only depend on starting condition and have no dependencies until the next trial. Also, the source data is not modified, and the state of the model is a relatively small data structure. This results in the ability to run multiple models simultaneously.

# REFERENCES

Print Publications:


[Abdelbar03]

Abdelbar, A.M. and M Mokhtar. "A k-elitist MAX-MIN ant system approach to cost-based abduction." Proceedings IEEE Congress onEvolutionary Computation (2003): 2635-2641.

[Abdelbar08]

Abdelbar, Ashraf . "Stubborn Ants." Swarm Intelligence Symposium (2008): 21-23.

[Belaid18]

Belaid, Ahror and Djamal Boukerroui. "Local maximum likelihood segmentation of echocardiographic images with Rayleigh distribution." Signal Image and Video (2018).

[Dice45]

Dice, R.L. "Measures of the Amount of Ecologic Association Between Species." Ecology (July 1945): Vol. 26, pp. 297-302.

[Dorigo04]

Dorigo, M and T St¨utzle. Ant Colony Optimization. Cambridge: MIT Press, 2004.

[Hadhoud12]

Hadhoud, MA and M Eladawy. "Left Ventricle Segmentation in Cardiac MRI Images." American Journal of Biomedical Engineering (2012): 131-135.

[Hadhoud11]

M. Hadhoud, M. Eladawy, A. Farag. "Automatic Global Localization of The Heart From Cine MRI images." IEEE International Symposium on IT in Medicine & Education (2011): 35-38.

[Jean-Yves10]
Jean-Yves, A, M Couprie and R Keriven. "Some links between extremum spanning forests, watersheds and min-cuts." Image and Vision Computing (2010): 1460–1471.

[Kaman10]
Kaman, M and K Selvanayaki. Improved implementation of brain MR image segmentation using Meta heuristic algorithms. Coimbatore, 2010.

[Martin04]
Martin, D. and J Fowlkers. "Learning to detect natural image boundariesusing local brightness, color, and texture cues." EEE Transactions on PatternAnalysis and Machine Intelligence (2004): 530-549.

[Masoumi12]
Masoumi, H and A Behrad. "Automatic liver segmentation in MRI images using an iterative watershed algorithm and artificial neural network." 2012.

[Monteiro09]
Monteiro, M and et al Fontes. "Ant Colony Optimization: a literature survey." 2009.

[Nadernejad08]
Nadernejad, Ehsan . "Edge Detection Techniques: Evaluations and Comparisons." Applied Mathematical Science (2008): Vol. 2, no. 31, 1507 - 1520 .

[Nageswararao17]
Nageswararao , AV and S Srinivasan. "Automatic hybrid ventricular segmentation of short-axis cardiac MRI images." Biomedical Researc (2017): 5816-5824.

[NCHS15]
CDC, NCHS. Underlying Cause of Death 1999-2013 on CDC WONDER Online Database. 2015 2 2015.

[Sobel14]
Sobel, Irwin. "An Isotropic 3x3 Image Gradient Operator." 2014.

[Storkey99]
Storkey, Amos and Romain Valabregue. "The basins of attraction of a new Hopfield learning rule." Neural Networks (1999): 869-876.

[Stutzle00]
Stutzle, T and H Hoos. "MAX- MIN Ant System." Future Generation Computer Systems 2000: vol.16, 889-914.

[Vermon91]
J.Vermon. "Strategies for graphical threshold determination." Computer Methods and Programs in Biomedicine (1991): 141-150.

[Zhao08]
Zhao, X and M Lee. Improved Image Thresholding Using ACO. DC: ACM, 2008.

[Zhuang04]
Zhuang, X. Edge feature extraction in digital images with the ant colony system. Boston: IEEE, 2004.

Appendix A

Sample Assignments

| Student # | Images #'s | | | Diagnostics |
|---|---|---|---|---|
| 1 | 39 | 17 | 25 | Ha: Sample Different? |
| | 30 | 19 | 26 | 25.67 mean |
| | 37 | 5 | 34 | 11.36 std.dev. |
| | 45 | 23 | 12 | 0.7 p-value |
| 2 | 17 | 25 | 35 | Ha: Sample Different? |
| | 19 | 26 | 48 | 26.42 mean |
| | 5 | 34 | 10 | 13.49 std.dev. |
| | 23 | 12 | 36 | 0.94 p-value |
| 4 | 25 | 35 | 44 | Ha: Sample Different? |
| | 26 | 48 | 8 | 27.33 mean |
| | 34 | 10 | 33 | 13.23 std.dev. |
| | 12 | 36 | 18 | 0.48 p-value |
| 4 | 35 | 44 | 24 | Ha: Sample Different? |
| | 48 | 8 | 7 | 24.33 mean |
| | 10 | 33 | 22 | 13.81 std.dev. |
| | 36 | 18 | 14 | 0.92 p-value |
| 5 | 44 | 24 | 28 | Ha: Sample Different? |
| | 8 | 7 | 6 | 20.75 mean |
| | 33 | 22 | 16 | 11.55 std.dev. |
| | 18 | 14 | 29 | 0.37 p-value |
| 6 | 24 | 28 | 47 | Ha: Sample Different? |
| | 7 | 6 | 21 | 24.83 mean |
| | 22 | 16 | 46 | 13.59 std.dev. |
| | 14 | 29 | 38 | 0.94 p-value |
| 7 | 28 | 47 | 13 | Ha: Sample Different? |

| | | | | | |
|---|---|---|---|---|---|
| | 6 | 21 | 2 | 24.67 | mean |
| | 16 | 46 | 9 | 15.84 | std.dev. |
| | 29 | 38 | 41 | 0.97 | p-value |
| 8 | 47 | 13 | 20 | Ha: Sample Different? | |
| | 21 | 2 | 32 | 27.25 | mean |
| | 46 | 9 | 15 | 15.76 | std.dev. |
| | 38 | 41 | 43 | 0.52 | p-value |
| 9 | 13 | 20 | 40 | Ha: Sample Different? | |
| | 2 | 32 | 42 | 21.83 | mean |
| | 9 | 15 | 4 | 16.77 | std.dev. |
| | 41 | 43 | 1 | 0.53 | p-value |
| 10 | 20 | 40 | 3 | Ha: Sample Different? | |
| | 32 | 42 | 11 | 22.42 | mean |
| | 15 | 4 | 27 | 15.58 | std.dev. |
| | 43 | 1 | 31 | 0.62 | p-value |
| 11 | 40 | 3 | 39 | Ha: Sample Different? | |
| | 42 | 11 | 30 | 25.83 | mean |
| | 4 | 27 | 37 | 16.54 | std.dev. |
| | 1 | 31 | 45 | 0.75 | p-value |
| 12 | 3 | 39 | 17 | Ha: Sample Different? | |
| | 11 | 30 | 19 | 23.92 | mean |
| | 27 | 37 | 5 | 13.4 | std.dev. |
| | 31 | 45 | 23 | 0.88 | p-value |

# Vita

Shannon Birchell has achieved both a B.S. in Computer Science and B.A. in Mathematical Statistics as well as maintaining electronics and software development certifications. He has worked with companies such as CSX Transportation and Gallo Winery to support integral priorities. Further, within these endeavors, he has had regular interactions with companies such as NVidia, Amazon and IBM to resolve technical issues or integrate new technologies. As a professional in the field of data science, he has worked rigorously with machine learning models. These models have been value producing and productionized for consumption in daily business activities. A high expertise in programing languages and architectural design has been displayed in knowledge domains like Java, Clojure and PLSQL as well as micro services and information architecture. As a Software Engineer, he made contributions to the enterprise and the public domain such as GitHub. The contributions range from R interpreter for Clojure and image processing tools. As a specialist, his focus has been given to derive uses for ACO in modern modeling by augmenting current progressive strategies.