

UNIVERSITÀ DEGLI STUDI DI  
TRIESTE

Dipartimento di Ingegneria e Architettura



Laurea Magistrale in Ingegneria  
Elettronica e Informatica

Lung nodule segmentation with  
region-based fast marching method

Laureando  
Marko Savic

Relatore  
Prof. Giovanni Ramponi

Correlatore  
Prof.ssa Weiwei Du

Anno Accademico 2019/2020

# Abstract

*Purpose:* When dealing with computed tomography (CT) volume data an accurate segmentation of lung nodules is of great importance to lung cancer analysis and diagnosis. However, due to the variety of lung nodules and the similarity of visual characteristics between nodules and their surroundings, a robust segmentation of nodules becomes a challenging problem. The main purpose of this study was to implement existing image processing techniques and numerical methods in order to propose an algorithm that can accurately segment lung nodules in CT images.

*Methods:* Firstly the input data is preprocessed and the annotated lung nodules are extracted. Two different datasets were used (LIDC and phantom data). In order to segment the nodules an algorithm based on the fast marching method was used. In this, multiple expanding fronts are created across the image resulting in its separation in regions with similar features. Then, to generate the overall nodule segmentation, a merging method is used that combines region growing with K-means.

*Results:* Experimental results show that the proposed technique can accurately segment nodules, especially in solid cases. Juxtapleural and juxtavascular cases are handled well. Performance for non-solid and cavitary nodules is good, but in some cases the segmentation fails. The segmentation algorithm was compared to an active contours implementation, with a satisfying outcome. The evaluation was performed with an objective method on the phantom dataset and with a subjective method on the LIDC dataset.

*Conclusions:* The region-based fast marching method can segment lung nodules in a robust and accurate manner. Using two different evaluation methods proved the usefulness of phantom data and the dissimilarities between an objective and subjective evaluation were helpful.

Keywords: Segmentation; Fast Marching Method; Lung Nodules; Computed Tomography; Lung Phantom.

# Sommario

*Scopo:* Quando si ha a che fare con i dati generati dalla tomografia computerizzata (TC), una accurata segmentazione dei noduli polmonari è di grande importanza per l'analisi e la diagnosi dei cancri polmonari. Tuttavia, a causa della varietà di noduli polmonari e della somiglianza delle caratteristiche visive tra i noduli e l'ambiente circostante, una robusta segmentazione dei noduli diventa un problema oneroso. Lo scopo principale di questo studio era quello di implementare delle tecniche di elaborazione delle immagini esistenti e metodi numerici al fine di proporre un algoritmo in grado di segmentare accuratamente i noduli polmonari nelle immagini TC.

*Metodi:* In primo luogo i dati di input vengono preelaborati e i noduli polmonari annotati vengono estratti, sono stati utilizzati due diversi dataset (LIDC e phantom). Per segmentare i noduli è stato utilizzato un algoritmo basato sul fast marching method, vengono creati più fronti di espansione sull'immagine con conseguente separazione in regioni con caratteristiche simili. Infine viene utilizzato un metodo di unione delle regioni per segmentare il nodulo, che combina un approccio di region growing con K-means.

*Risultati:* I risultati sperimentali mostrano che la tecnica proposta può segmentare accuratamente i noduli, specialmente nei casi solidi. I casi juxtapleurici e iuxtavascolari sono gestiti bene. Le prestazioni per noduli non solidi e cavitari sono buone, ma in alcuni casi la segmentazione fallisce. L'algoritmo di segmentazione è stato confrontato con un'implementazione di active contours, dando dei risultati soddisfacenti. La valutazione è stata eseguita con un metodo oggettivo sul dataset phantom e con un metodo soggettivo sul dataset LIDC.

*Conclusioni:* Il region-based fast marching riesce a segmentare i noduli polmonari in modo robusto e accurato. L'utilizzo di due diversi metodi di valutazione ha dimostrato l'utilità dei dati phantom e le differenze tra una valutazione oggettiva e soggettiva sono state fruttuose.

Parole chiave: Segmentazione; Fast Marching Method; Noduli Polmonari; Tomografia Computerizzata; Phantom Polmonare.

# Contents

<b>Abstract</b>	i
<b>Sommario</b>	ii
<b>Introduction</b>	v
Objective . . . . .	v
General . . . . .	v
State of the art . . . . .	vi
Credits and original work . . . . .	viii
Novel contributions . . . . .	ix
Structure . . . . .	x
<b>1 Input and preprocessing</b>	1
1.1 Input data . . . . .	2
1.1.1 Computed Tomography . . . . .	2
1.1.2 Input datasets . . . . .	3
1.1.3 Loading the DICOM files . . . . .	4
1.2 Resampling and filtering . . . . .	6
1.3 Segmentation of the lungs' area . . . . .	8
1.4 Normalisation and cropping . . . . .	11
<b>2 Lung nodule segmentation</b>	12
2.1 FMM background . . . . .	13
2.2 FMM algorithm . . . . .	16
2.2.1 Algorithm outline . . . . .	16
2.2.2 Algorithm implementation . . . . .	20
2.2.2.1 Status . . . . .	20
2.2.2.2 Times . . . . .	22
2.2.2.3 Heap and Region . . . . .	23
2.2.2.4 Implementation . . . . .	24
2.3 Segmentation - preparation . . . . .	26
2.3.1 Speed function . . . . .	26
2.3.2 Seed grid . . . . .	28
2.4 Segmentation - region merging . . . . .	30

2.4.1	K-means clustering . . . . .	31
2.4.2	Merging . . . . .	33
2.4.3	Parameter tuning . . . . .	35
<b>3</b>	<b>Evaluation</b>	<b>37</b>
3.1	Comparison method . . . . .	38
3.2	Subjective evaluation . . . . .	39
3.2.1	Dataset construction . . . . .	39
3.2.2	Evaluation method . . . . .	41
3.2.3	Results . . . . .	43
3.2.3.1	Round . . . . .	44
3.2.3.2	Irregular . . . . .	49
3.2.3.3	Semi-transparent . . . . .	54
3.2.3.4	Cavitory . . . . .	60
3.3	Objective evaluation . . . . .	64
3.3.1	Dataset construction . . . . .	64
3.3.2	Evaluation method . . . . .	69
3.3.3	Results . . . . .	73
3.3.3.1	Round . . . . .	74
3.3.3.2	Irregular . . . . .	81
3.3.3.3	Semi-transparent . . . . .	87
3.3.3.4	Cavitory . . . . .	93
3.4	Discussion and comparison . . . . .	100
<b>Conclusions</b>		<b>102</b>
<b>A</b>	<b>Segmentation results - LIDC</b>	<b>103</b>
A.1	Round . . . . .	104
A.2	Irregular . . . . .	112
A.3	Semi-transparent . . . . .	121
A.4	Cavitory . . . . .	129
<b>B</b>	<b>Segmentation results - Phantom</b>	<b>133</b>
B.1	Round . . . . .	134
B.2	Irregular . . . . .	146
B.3	Semi-transparent . . . . .	158
B.4	Cavitory . . . . .	164

# Introduction

## Objective

The present master thesis is the result of a collaboration between the Kyoto Institute of Technology and the University of Trieste. Most of the work was performed while the author was a intern at the Visual Information Lab at KIT, under the supervision of professor Weiwei Du.

When dealing with computed tomography (CT) volume data an accurate segmentation of lung nodules is of paramount importance to lung cancer analysis and diagnosis. The main purpose of this study was to implement existing image processing techniques and numerical methods in order to propose an algorithm that can accurately segment lung nodules in CT images. This work implements a region-based extension to the fast marching method (FMM) to segment lung nodules. The results are evaluated by applying different methods on both phantom and real patient data.

## General

Lung cancer is one of the most malignant tumors in the world and its 5-year-survival rate is only 18% [1]. The use of CT images for treatment, monitoring, and analysis is an important strategy for early lung cancer diagnosis and survival time improvement [2]. The demonstrated 20% reduction in lung cancer specific mortality in the National Lung Screening Trial (NLST), and the consequent adoption of lung cancer screening has emphasized the importance of nodule detection and evaluation [3, 4]. With the development of CT technology, images can be generated in mere seconds, a single scan contains at least a few hundred images. Each examination generates images in the order of hundreds that radiologists have to analyze. This can be very challenging work because radiologists must read a large number of images and detect lung nodules without mistakes, it can be a big physical and mental burden. Moreover, even if the lung nodule is detected, in some cases it is difficult to judge whether they are benign or malignant, because there are not enough clear features to form an accurate classification.

To determine malignancy the surest approach is to monitor the lung nodule over time and observe whether there have been significant changes in volume. Thus, accurate segmentation is of great significance in lung nodule analysis.

## State of the art

In view of the exceptional importance of nodule segmentation, several efforts have been made to develop a highly accurate and robust automatic nodule segmentation system that may assist radiologists. Achieving accurate results is still very challenging because of the many different features lung nodules possess. These efforts can be distinguished into deep learning-based techniques and classical image processing-based techniques.

Numerous methods based on the morphological operation were proposed for lung nodule segmentation, Kostis et al. [5] applied morphological operations in order to remove the nodule-attached vessels, and lung nodules were then isolated according to the selection of connected regions. Even though these methods are fast and easy to implement, the size of the morphological operator is difficult to set due to the various sizes of the nodules. Kuhnigk et al. [6] carried out a more complex morphological correction which allows to manage both small and big size nodules, that usually have wide contact surfaces with other anatomical objects.

In region growing methods, segmentation starts with a user-specified seed point, and voxels are included into a nodule set iteratively until the predefined convergence criterion is satisfied. These methods work well for isolated nodules, however when analyzing juxtapleural nodules, region growing algorithms are known to be difficult to converge, due to the similarity between juxtapleural nodules and surrounding regions [7]. To alleviate this problem, Dehmeshki et al. proposed a new region growing method based on intensity information, distance, fuzzy connectivity and peripheral contrast [8]. Although they introduced a variety of rules which are still not adapted well to irregularly shaped nodules. The biggest challenge in region growing methods is the convergence criteria which is difficult to set because of the wide variety and irregular shapes of nodules.

In energy optimization methods, nodule segmentation is converted into an energy minimization task. The level set based methods, for example, use a level set function to describe the image, and the function is minimized when the segmented contour matches the nodule boundary [9]. To detach nodules from lung wall, Farag et al. combined level set with shape prior hypothesis [10]. Boykov and Kolmogorov et al. [11] used a graph cut method for lung nodule segmentation by framing the problem into a maximum flow optimization task. The energy optimization methods perform well in case of isolated nodules but are not well adapted to ground glass opacity (GGO) nodules and juxtapleural nodules.

There has been research regarding GGO lung nodule segmentation with expectation–maximization algorithm [12] and active contour models [13], where for part-solid nodules the solid and non-solid parts are segmented separately and combined.

In recent years, an increasing number of studies have developed artificial intelligence tools in the field of medical image segmentation.

Lu et al. have proposed a stratified learning framework including (supervised) object-specific image segmentation, segmentation feature extraction, robust object classification and importance regression [14]. Hu et al. performed segmentation of the lungs’ area and next carried out a Hessian matrix-based vascular feature extraction procedure to obtain the lung blood vessel mask. Using the difference between the two masks, 3D regions are extracted representing candidate pulmonary nodules, subsequently classifying them with the aid of a neural network [15]. Gonçalves et al. developed a different approach that uses Hessian-based strategies for lung nodule segmentation. They propose a multiscale segmentation process that uses the central medialness adaptive principle, a Hessian-based strategy that was originally formulated for tubular extraction but it also provides good segmentation results in blob-like structures as is the case of lung nodules [16]. The method proposed by Jung et al.’s can separate solid component and ground-glass opacity using an asymmetric multi-phase deformable model [17].

Focussing on the deep learning approach, several algorithms have been proposed recently. A convolutional neural network (CNN) is a multi-layered neural network which learns to map original image files and corresponding labels hierarchically, and the task of segmentation is modified into the classification of voxels similar to that used in the previous machine learning operations. For example, Wang et al. introduced the multi-view convolutional neural network (MVCNN) for nodule segmentation, which specialized in capturing a diverse set of nodule-sensitive features from axial, coronal and sagittal views in CT images simultaneously [18]. On the other hand, Fully convolutional networks FCN [19] were a different method used for segmentation. Ronneberger et al. proposed 2D U-Net architecture with a segmentation approach which is better-adapted to biomedical imaging.

Recently, variations of CNN, like the central focused convolutional neural network (CF-CNN) proposed by Wang et al. [20], and the Dual-branch residual network (DBResNet) for lung nodule segmentation proposed by Cao et al. [21], have achieved competitive segmentation performance.

## Credits and original work

In this section the ideas, techniques, algorithms and code, which were used for this work are listed. Some of these are original and some built upon existing work as a foundation. Most of the plots and figures were created with the help of the Matplotlib library [22].

- Preprocessing : for this part, several techniques and code were used from two separate solutions to the preprocessing part of the Lung Nodule Analysis 2016 (LUNA16) challenge [23] [24].
  - Loading the DICOM files, resampling and normalising the final results were performed using code from [23].
  - Filtering the image in the frequency domain was an original addition and the filter was designed and implemented from scratch.
  - Segmentation of the lungs' area was performed using and modifying the code from [24], with the substitution of the binary closure step with the convex hull, giving better results in areas where there are juxtapleural nodules.
  - Importing the annotations and cropping the nodules was done with original code.
- Fast Marching Method : for this part most of the code was written from the ground up, except for the use of some functions from Python image processing libraries. Several techniques were designed taking inspiration from already existing work.
  - For the FMM preparation, the smoothing and gradient magnitude were performed with functions from the SimpleITK library [25]. The seed grid generation method draws heavily from [26], where an automatic initialisation for the seeds is proposed based on statistical analysis. A similar method was designed that makes use of the image gradient and uses a kindred shifting technique, the implementation was done with original code. The padding scheme which improves performance for juxtapleural nodules was an original idea.
  - FMM was implemented by writing the code completely from the beginning, by following the algorithm as described by J.A. Sethian in [27]. For the addition of region assigning to multiple fronts the idea from [28] was implemented, where a new region growing method is proposed for segmenting images using a multi-label fast marching algorithm.

- The scikit-learn Python library [29] library was used to execute K-means, and cluster merging was carried with an often used criteria for region growing, for which the recursive algorithm was implemented from the ground up. The trackbars were added using functions from the matplotlib library.
- Evaluation : for this part most of the code was written from the ground up, except for the use of some basic functions from Python image processing libraries, and for the Matlab [30] implementation of active contours. Several techniques were designed taking inspiration from already existing concepts.
  - For the comparison method the *activecontours* function was used from Matlab as is, without making any significant modifications.
  - For the subjective evaluation the questionnaires were written from scratch keeping in mind that the radiologists task and the scoring system should be well defined. To extract the data from the completed forms the PyPDF2 Python toolkit [31] was used.
  - For the objective evaluation the phantom dataset was generated by using image processing tools from GIMP. Implementing the evaluation metrics was done writing original code by following the definitions of both IoU and long-axis/short-axis diameter measure.

## Novel contributions

This work was based and built upon existing image processing techniques, with the following original contributions:

- Improving existing preprocessing methods by adding adequate filtering and refining segmentation of the lungs' area. This way removing unwanted high frequency artifacts and bettering segmentation for juxtapleural nodules.
- Modifying the existing automatic seed grid generation method, making it more suitable for lung nodule segmentation. By utilizing the gradient mean as threshold and shifting seeds diagonally as well.
- Devising and applying a padding scheme to improve segmentation for juxtapleural nodules, countering the undesirable effect the lung wall has on the speed function.

- The region-based method was inspired by existing multi-label fast marching method implementations, with the original addition of using K-Means to cluster the regions before merging, resulting in a more robust segmentation without having to sacrifice detail. Furthermore, the method for selecting the first cluster to be merged was original, and it combines distance from the center and mean intensity to make the decision.
- Evaluating the performance with both a subjective and objective method for comparison. Where for the objective method an "artificial" nodules dataset was created. This way the evaluation can be done in greater depth, recognizing possible biases each method can have.

## Structure

In the first chapter the techniques which were used for preprocessing are described, starting from the raw CT data and ending with an output of appropriately sized images centered on the lung nodules which will be inspected.

The second chapter illustrates the algorithm used for lung nodule segmentation. Beginning from an outline and description of the fast marching method, then moving on to details about its implementation and the necessary extensions made to provide accurate segmentation results.

These results are subsequently evaluated as described in chapter three, using two fundamentally different methods applied on two different datasets. The outcomes are compared and discussed.

The thesis is concluded by some final considerations summarising the work done. After the conclusions, all segmentation instances are shown in the two appendices.

# Chapter 1

## Input and preprocessing

Before proceeding with the segmentation algorithm the raw data provided by a Computed Tomography (CT) scanner needs to be processed with the purpose of rendering it more meaningful and useful. This chapter deals with explaining the processes that were utilized in order to prepare the data for the segmentation algorithm, it will give a detailed explanation of how the data created by a CT scanner was transformed in images that are ready to be inputted in the segmentation algorithm. The input of this preprocessing segment is a .dcm file representing the CT volume data of a scan and the output is a set of appropriately sized images centered on lung nodules. The nodule centroids were found using annotations for LIDC data and manually for the phantom data.

The methods discussed in this chapter were implemented using Python, the main reasons behind this choice were the fact that it is open source, it has extensive libraries especially for image processing and is a very versatile language. Various methods and code were taken from two separate solutions to the preprocessing part of the LUNA16 challenge [23] [24].

The preprocessing stage consists of the following steps:

1. Input data : this first step consists of transforming the data obtained from a CT scanner into a more manageable format. The slice images and parameters are extracted from the .dcm files and the pixel intensity values are transformed to fit the Hounsfield scale.
2. Resampling and filtering : to render the data more homogeneous, all slices were resampled to the same spacing, and an adequate filter was applied to avoid imaging artifacts.
3. Segmentation of the lungs' area : to remove all the parts in the image that are of no interest, a segmentation technique is used that keeps only the lung tissue and eliminates everything else from the image.
4. Normalisation and cropping : finally the resulting images are normalised and cropped so that they are centered on lung nodules.

## 1.1 Input data

The input data used for this study consists of two separate datasets containing CT scans, the purpose of having both datasets will be explained thoroughly during the evaluation phase.

### 1.1.1 Computed Tomography

Computed Tomography is a medical imaging procedure that was first introduced more than 40 years ago and has since become an integral part of clinical practice [32]. In first generation (conventional) scanners, an electronic tube produces a narrow beam of X-rays that passes through the patient and is picked up by a row of detectors on the other side. The tube and detectors are positioned on opposite sides of a ring that rotates around the patient [33]. Modern scanners have evolved to be more efficient and accurate, but the core principle remains the same, many X-ray measurements are taken from different angles and combined to produce cross-sectional images of specific areas of a patients body. Compared to X-ray radiography (2D X-ray) a CT scan produces a 3D volume with higher resolution images, which provide a much more detailed structure of the affected body part and clearer images of the internal soft tissues and organs, the only disadvantages are the higher cost and longer examination times. CT scanners have become fundamental in identifying and assessing lung nodules.

Out of the many parameters [34] that characterize a specific scan, the most important that will be considered are:

- Tube current : increases in tube current or the product of tube current and scan time (mAs) result in improved image quality, decreased image noise and increased patient dose.
- kVp : as a general rule of thumb, the radiation dose changes with the square of kVp, decreases in kVp can result in nonlinear exponential increases in image noise often necessitating a concomitant increase in milliampere-seconds to preserve image quality.
- Slice thickness : represents the axial resolution. As the slice thickness decreases the number of photons within each voxel also decreases resulting in increased image noise.
- Pixel size : represents the spatial resolution of a slice, it is given as the ratio between the field of view (which is the maximum diameter of the area of the scanned object that is represented in the reconstructed image, measured in mm ) and the image matrix size (typically 512x512).

### 1.1.2 Input datasets

The input data was taken from two different sources. The first dataset was specifically generated for this study by Tianjin chest hospital, the CT volume data was produced by scanning a lung phantom containing spherical nodules with different sizes. In Figure 1.1 the lung phantom is displayed together with a single axial slice containing nodules.

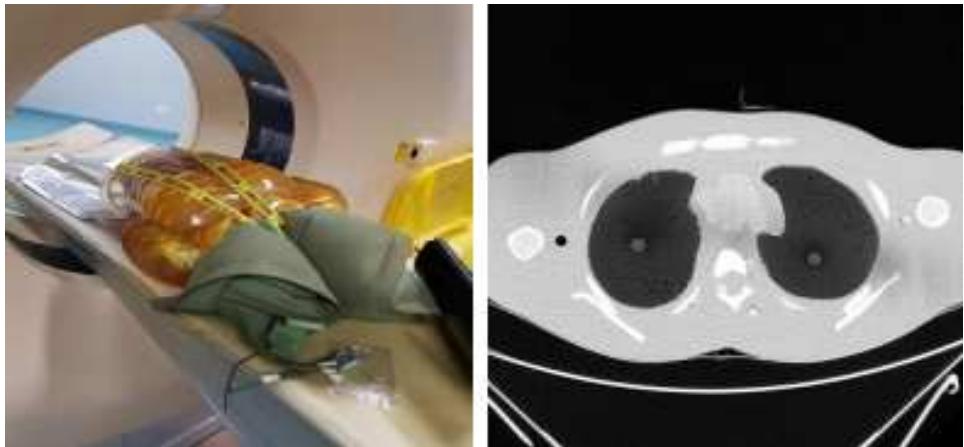


Figure 1.1: Lung phantom placed in CT scanner and single axial slice

The second dataset used for experimentation was taken from the Lung Image Database Consortium (LIDC) database [35], those scans were acquired from multi-detector row CT scanners with a wide range of scanning parameters [36]. The values or ranges from the two datasets are shown in Table 1.1, for the phantom scans the only parameter that changes between scans is the tube current ( higher tube currents result in better image quality and reduced noise ). Conversely the LIDC dataset contains a great number of different scans and the parameter range is broad, providing good variety.

	Phantom	LIDC
Tube current [mA]	30 - 197	40 - 582 (average 177)
Kilovoltage peak [kVp]	120	120, 130, 135, 140
Slice thickness [mm]	0.625	0.625 - 3.000 (average 1.7)
Pixel size [mm]	0.683	0.508 - 0.946 (average 0.66)

Table 1.1: CT scan parameters

Due to the heterogeneous nature of the data the preprocessing step is of great importance.

### 1.1.3 Loading the DICOM files

DICOM (Digital Imaging and Communications in Medicine) is a standard for handling, storing, printing and transmitting information in medical imaging, it includes a file format definition and a network communications protocol. A DICOM data object consists of a number of attributes or tags containing information and parameters of the specific scan, and also one special attribute containing the image pixel data ( in CT scans it is usually 512x512 pixels). For every slice of the scan there is a corresponding DICOM file. Loading the files was performed using the pydicom [37] library, which is a pure Python package for working with these types of files, it was made for inspecting and modifying DICOM data in an easy "Pythonic" way. More specifically, each scan contains hundreds of slices, and every slice is associated with a .dcm file which is loaded as a Python object. These new objects inherit all the attributes from the slices, of which the image pixel data is saved as a NumPy[38] array. NumPy is a Python library which adds support for arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Since the nodule locations are known beforehand, only the slices containing lung nodules go through further processing, the rest of the slices discarded. Each of the following preprocessing steps is applied to slice by slice. After loading the image pixel data and relevant attributes, it is necessary to convert from the 12 bit grayscale values found in CT data to Hounsfield units (HU) [39]. This scale, named after Sir Godfrey Newbold Hounsfield the inventor of CT scanning, is a quantitative measure of radiodensity used in evaluating CT scans. Attenuation values are expressed according to this linear density scale, in the Hounsfield scale water is arbitrarily assigned a value of 0 HU and air the value of -1000 HU. In Table 1.2 the typical values for different substances are shown.

Substance	HU
Air	-1000
Lung	-700
Soft Tissue	-300 to -100
Fat	-84
Water	0
CSF	15
Blood	+30 to +45
Muscle	+40
Bone	+700 to +3000

Table 1.2: Hounsfield units for different substances

The grayscale intensity of each pixel from the DICOM file needs to be remapped with a linear function, which is shown in equation 1.1.1.

$$HU = \text{pixel\_value} \cdot \text{RescaleSlope} + \text{RescaleIntercept} \quad (1.1.1)$$

Where RescaleSlope and RescaleIntercept are two parameters that can be found in the related DICOM tags, they are determined by the manufacturer of the hardware and typically CT images have a RescaleIntercept of -1024 and RescaleSlope of 1.

In Figure 1.2 a slice is displayed with corresponding histogram in HU, the highest frequency is around -1000 HU which corresponds to air, the second peak corresponds to the lung parenchyma which is around -700 HU and the third peak corresponds to water and soft tissue which are around 0 HU.

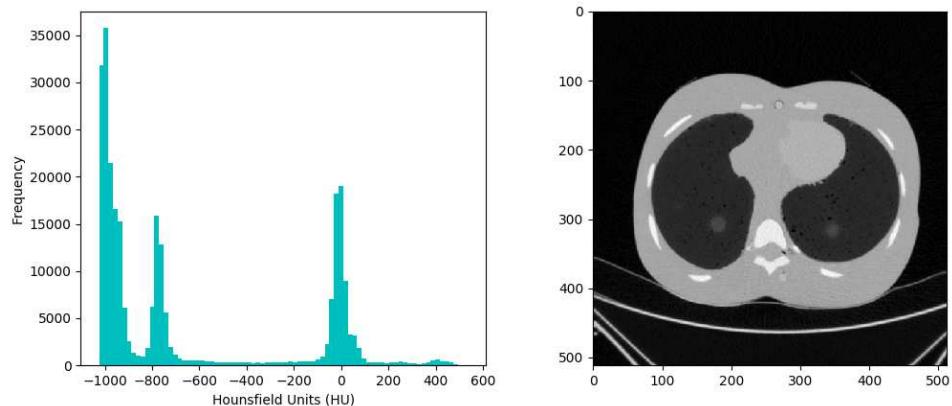


Figure 1.2: Slice with HU histogram

## 1.2 Resampling and filtering

When dealing with CT volume data the pixel spacing changes from scan to scan, it is more convenient to resample all scans from the database to the same pixel spacing, rendering the data more homogeneous. The selected value corresponds to  $0.5 \times 0.5 \text{ mm}^2/\text{pixel}$ , this spacing was chosen to avoid loss of details in the images, since it is lower than the minimum spacing of the whole dataset, thus resulting in images with slightly higher resolution. Reducing the resolution would be counter productive considering that details are already lacking, especially for small nodules. This resampling was performed by using spline interpolation of the third order, specifically with the `scipy.ndimage.interpolation.zoom` function from the SciPy Python library [40]. The resize factor is equal to the original pixel spacing divided by the desired value of 0.5. This operation will add some undesired higher frequency components, which will be mitigated by an appropriate filter. The result is filtered with a two-dimensional lowpass Butterworth filter [41] to avoid these imaging artifacts, this filter was chosen because it provides a maximally flat magnitude response in the pass-band.

The filter is applied in the frequency domain to each slice of the scan making use of the convolution property of the Fourier transform [42], which states that under suitable conditions the Fourier transform of a convolution of two signals is the pointwise product of their Fourier transforms.

Let  $f$  and  $g$  be two functions,  $f * g$  their convolution and  $\mathcal{F}$  the Fourier transform operator, then equation 1.2.1 describes this property:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\} \quad (1.2.1)$$

Each slice image is taken to the frequency domain using a fast Fourier transform (FFT), then multiplied by the frequency response of the filter and subsequently taken back to the spatial domain with a inverse fast Fourier transform (IFFT), these operation are performed using functions from the `numpy.fft` module. The lowpass Butterworth filter has been designed by utilizing equation 1.2.2:

$$G(\omega) = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (1.2.2)$$

Since it is a two-dimensional filter the spatial frequency  $\omega$  is seen as the distance from the center in the spatial frequency plane,  $G(\omega)$  is the gain, n is the order (  $n = 4$  has been chosen for a sufficiently steep roll-off ), and  $\omega_c$  the cut-off frequency (  $\omega_c = \text{resizefactor}^{-1}/2$  ).

In Figure 1.3 is an example of how this process can affect the frequency spectrum. A slice from a CT scan was taken with magnitude spectrum displayed in (a), after resampling the resulting magnitude spectrum is (b) which has a wider band and undesired high frequency components (which are highlighted in red). The frequency response of the 4th order lowpass Butterworth filter can be seen in (c), the response within the pass-band is very flat. The filter is applied resulting in the magnitude spectrum seen in (d), where the high frequency components are attenuated.

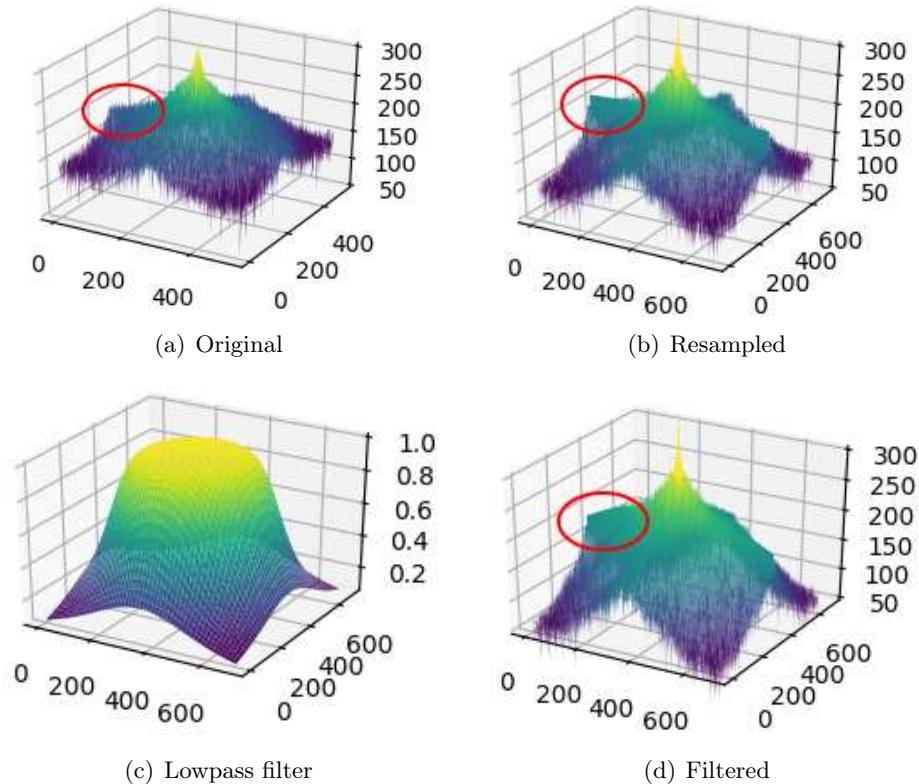


Figure 1.3: Effects of resampling and filtering on spectrum

### 1.3 Segmentation of the lungs' area

To remove all the undesired elements from the slice image and only keep the lung tissue, a segmentation method is used based on connected component analysis and morphological operations. This helps reduce the problem space and is an important step for juxtapleural nodules, this way they are separated from the lung wall and will need less special considerations going forward. Most of the image processing tools used for this stage are taken from the scikit-image Python library [43]. The lung segmentation is done via the following steps, which can be seen in Figure 1.4:

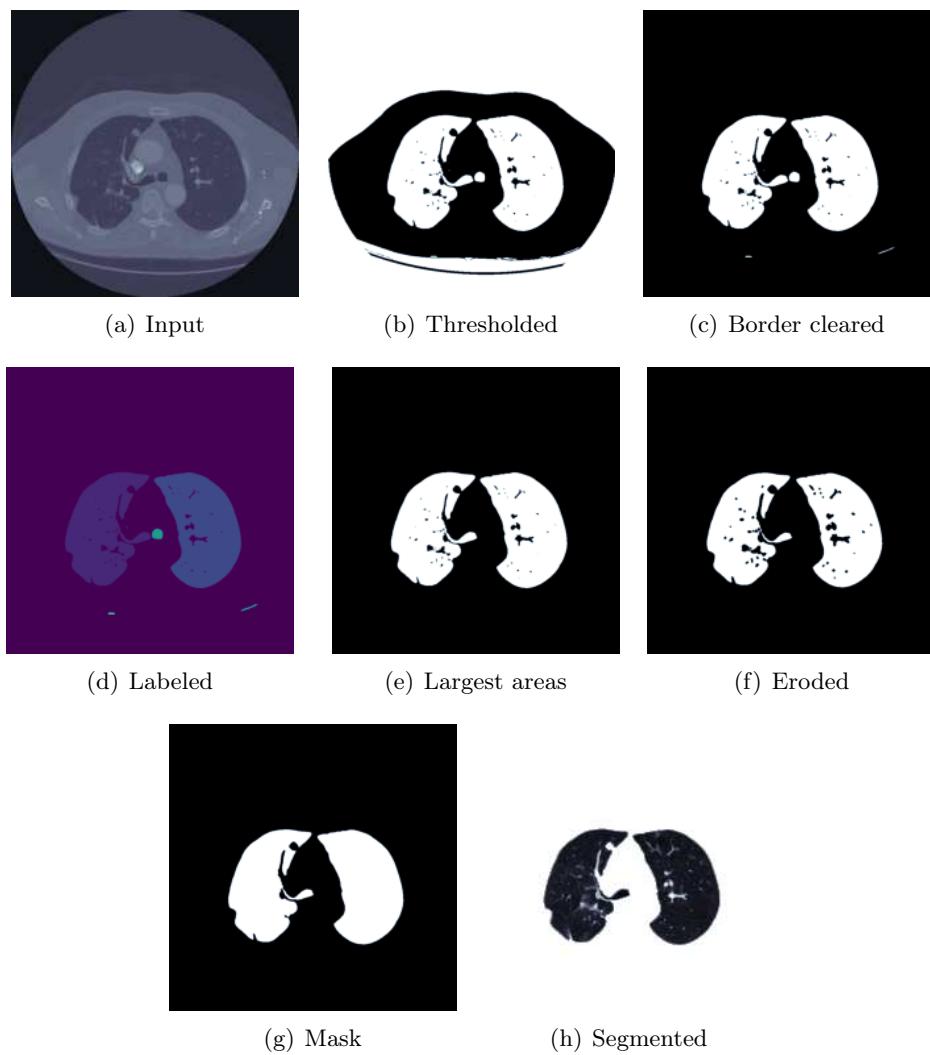
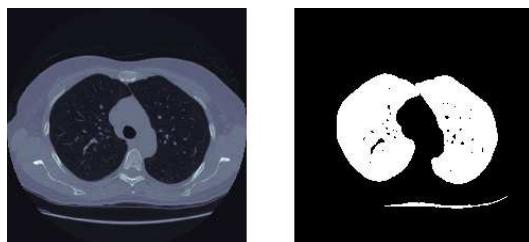


Figure 1.4: Segmentation of the lungs' area example

- Firstly a binary image is created by thresholding the slice image at -400 HU, this eliminates all values above the threshold which correspond to soft tissue and bone, leaving the lung tissue and air intact.
- The area outside of the patients body is made up of mostly air and needs to be removed from the binary image. The best way to do this is by using a function that clears objects connected to the label image border, which is called `skimage.segmentation.clear_border`.
- Each connected component in the binary image is assigned an integer label by utilizing another function called `skimage.measure.label`
- To make sure only the lungs remain, the two connected components with the largest area are kept. This is implemented via the function `skimage.measure.regionprops` that measures properties of labeled image regions, one of these properties is the area. Having the area of each labeled region selecting the two connected components with the largest area is immediate. In some edge cases the lungs correspond to only one connected area, this happens when only one lung is visible on the slice or when the lung walls are touching, a few examples are shown in Figure 1.5. The resulting mask won't give a perfect segmentation of the lungs, but it doesn't have a negative impact because the superfluous area ( typically the trachea or something outside the patients body ) in the image is far enough from the lungs that it won't show up in the final cropped nodule images.



(a) Lungs are connected



(b) Only one lung is visible

Figure 1.5: Edge cases with two largest connected components

- A couple of morphological operation are applied to improve the mask. At first erosion with a disk of radius 2 is applied to separate the lung nodules attached to the blood vessels, with the function *skimage.morphology.binary\_erosion*. Subsequently *scipy.ndimage.morphology.binary\_fill\_holes* is used to fill the small holes in the mask, this function consists in invading the complementary of the shapes in input from the outer boundary of the image, using binary dilations. Holes are not connected to the boundary and are therefore not invaded. The result is the complementary subset of the invaded region.
- Finally the mask is applied to the original slice image to obtain the segmented lung.

This method does not yet take into account juxtapleural nodules, in fact it produces a binary mask that excludes them because they are connected to the outside tissue. This can be resolved with binary closing, but it has an unwanted effect on the binary mask when the connecting area is large, because it requires a large kernel which has the unwanted effect of removing significant parts of the image. A better way of dealing with this issue is by exploiting convex hulls, the convex hull of a binary image is the set of pixels included in the smallest convex polygon that surround all white pixels in the input. At the next stage the image is cropped to a smaller one which is centered on the nodule, and in case of juxtapleural nodules it will result in a hollow middle, this can be detected by checking if the intensity value at the central pixel is zero. When the anomaly is detected a binary image is created by thresholding at above zero intensity, afterwards the function *skimage.morphology.convex\_hull\_image* is applied on the binary image resulting in a new mask where the hole is covered. Because the lung wall generally has a convex shape this solution works better in preserving juxtapleural nodules, and gives a more accurate result independent of the size of the cavity. The effects can be seen in Figure 1.6. In case of juxtapleural nodules, the lung segmentation fails and leaves a cavity instead of the nodule (b). This cavity can be filled by using binary closing (c), but using a binary mask based on the convex hull gives better results.

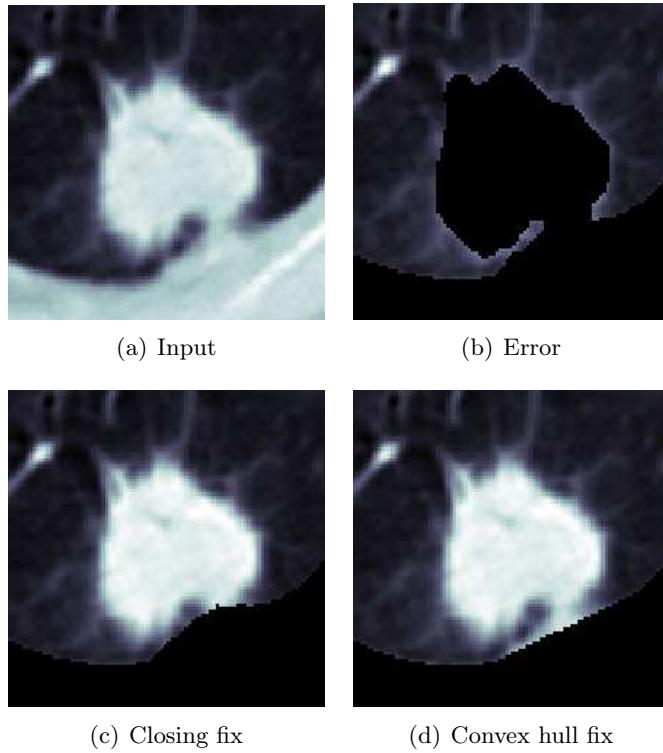


Figure 1.6: Juxtapleural fix

## 1.4 Normalisation and cropping

After lung segmentation the image is normalised to fit the range of -1000 HU to +400 HU, anything outside of this range is of no interest ( bones with different radiodensity ).

The final step is cropping the images to a smaller and more manageable size, working on the whole slice doesn't make sense because apart from the nodule and a small area outside its boundary the rest of the slice image is irrelevant to the segmentation and would only slow it down considerably.

The size was picked based on the nodule diameter, the height and width of the cropped image are twice the diameter, this way the nodule is included with enough surrounding tissue. For the phantom dataset the nodule diameter corresponds to the actual diameter of the spheres, and the approximate centroid coordinates were annotated by the author. For the LIDC dataset the diameter and centroid were taken from the annotations provided for the LUNA16 Kaggle challenge, this data is easier to handle than the data that is provided directly with the LIDC dataset because the centroid is already calculated. The end result, which will be used as input for the segmentation algorithm, is a set of centered nodule images.

## Chapter 2

# Lung nodule segmentation

The segmentation method is based on using the fast marching method (FMM) with multiple seed points to produce a set of regions in the image that will be merged accordingly in order to segment the nodule. This method was chosen as the core of the segmentation algorithm due to its efficiency and accuracy. Several applications of the fast marching method in image segmentation were proposed, a multi-label version of the algorithm was introduced [28] and also a generalised method [44]. Specific applications in medical image processing have yielded successful results, in [45] FMM was used for automatic liver segmentation in CT images. As with the previous chapter, the implementation of the methods that will be discussed was done with Python, with most of the code written from the ground up. Several techniques were inspired from existing work. The main implementation was performed by following the algorithm as described by J.A. Sethian in [27], and by adding a region assigning component inspired by previous work [28]. The seed grid generation method draws heavily from [26].

The segmentation method is divided in three parts:

- Preparation : with the nodule image as input, a grid of starting points (or seeds) and a speed function are created. Both these elements are necessary for FMM.
- Fast marching method implementation : the method is applied iteratively until an arrival time is calculated for every pixel and every pixel is assigned to a region. The end results are two matrices, the first containing all the arrival times and the second contains labeled regions.
- Region merging segmentation : finally the regions are merged using a region growing method and the nodule is segmented.

At first some background is provided, proceeding with the FMM algorithm outline and implementation, followed by preparation and region merging segmentation.

## 2.1 FMM background

The fast marching method [27] is an algorithm that was developed with the goal to solve a discretised version of the Eikonal equation on a uniformly sized spatial grid. The Eikonal equation is a non-linear partial differential equation encountered in problems of wave propagation, it is derivable from Maxwell's equations of electromagnetics. Its formulation is shown in equation 2.1.1

$$\begin{cases} |\nabla u(x)|f(x) = 1 & \forall x \in \Omega \subset \mathbb{R}^n \\ u(x) = q(x) & \forall x \in \partial\Omega \end{cases} \quad (2.1.1)$$

It arises naturally in many applications including continuous optimal path planning, computational geometry, photolithography, optics, shape from shading and image processing. One natural interpretation for the solution of equation 2.1.1 comes from isotropic time-optimal control problems. For a vehicle traveling through  $\bar{\Omega}$ ,  $f$  describes the speed of travel and  $q$  gives the exit time-penalty charged on  $\partial\Omega$ . In this framework,  $u(x)$  is the value function, i.e., the minimum time to exit  $\bar{\Omega}$  through  $\partial\Omega$  if we start from a point  $x \in \Omega$  [46]. In practice, firstly a speed function is defined for each point in the spatial grid and a starting point is chosen, from which arrival times are calculated iteratively for each point in the grid by using the speed function, the result obtained is a map of arrival times.

The algorithm is similar to Dijkstra's algorithm [47] (meant for finding the shortest paths between nodes in a graph) and uses the fact that information only flows outward from the seeding area. This problem is a special case of level set methods.

Level set method [48] is a numerical solution for processing topological changes of contours, this algorithm has been widely used in the field of image processing especially in image segmentation. The basic idea is to represent the curves or surfaces as the zero level set of a higher dimensional hyper-surface. This technique not only provides more accurate numerical implementations but can also handle topological change very easily. Essentially, it means that the closed curves in a two-dimensional surface are regarded as a continuous surface of a three-dimensional space. Thus, the evolution of a curve can be transformed into the evolution of a three-dimensional level set function.

Given an initial position for an interface  $\Gamma$ , where  $\Gamma$  is a closed curve in  $\mathbb{R}^2$ , and a speed function  $F$ , which gives the speed of  $\Gamma$  in its normal direction, the level set method takes the perspective of viewing  $\Gamma$  as the zero level set of a function  $\phi(x, t = 0) : \mathbb{R}^2 \rightarrow \mathbb{R}$ . That is, let  $\phi(x, t = 0) = \pm d$ , where  $d$  is the distance from  $x$  to  $\Gamma$ , and the plus (minus) sign is chosen if the point  $x$  is outside (inside) the initial hyper-surface  $\Gamma$ . Then, by the chain rule, an evolution equation for the interface may be produced, as shown in equation 2.1.2

$$\begin{aligned} \phi_t + F|\nabla\phi| &= 0, \\ \text{given } \phi(x, t = 0) \end{aligned} \tag{2.1.2}$$

This is an initial value partial differential equation in one higher dimension than the original problem [27]. This equation describes the time evolution of the level surface function  $\phi$  in such a way that the zero level set of this evolving function is always identified with the propagating curve.

In Figure 2.1 is an example of a propagating circular front, showing the outward propagation of the initial curve and the accompanying motion of the level set function  $\phi$ .

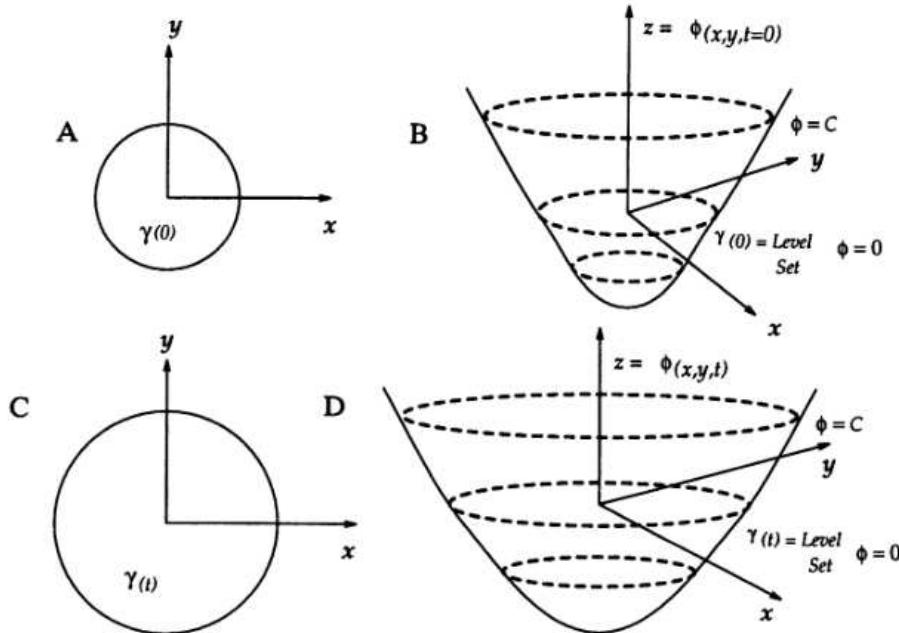


Figure 2.1: Propagating circle [27]

A disadvantage of this method is that it is computationally expensive, since an extra dimension has been added, and its continuous nature is not fit for discretized problems. A more efficient discrete application is the fast marching level set method. Let's imagine a two dimensional case with a front moving with speed  $F(x, y) > 0$ , let's suppose that for the propagating curve we graph the evolving zero level set above the xy plane. That is, let  $T(x,y)$  be the time at which the curve crosses the point  $(x,y)$ . Then the surface  $T(x,y)$  satisfies equation 2.1.3.

$$|\nabla T|F = 1 \quad (2.1.3)$$

Meaning that the gradient of the arrival time surface is inversely proportional to the speed of the front, this is a form of the Eikonal equation. Considering a uniform mesh, denoting by  $h$  the space step in both directions and by  $T_{i,j}$  the approximation of  $T$  at node  $(x_i, y_j) = (ih, jh)$ , to approximate the gradient of  $T$  equation 2.1.4 is used [49].  $D_{ij}^{+x}T$  and  $D_{ij}^{-x}T$  are spatial derivative operators.

$$|\nabla T|_{ij} \approx [max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2]^{\frac{1}{2}}$$

$$\text{with } D_{ij}^{-x}T = \frac{T_{i,j} - T_{i-1,j}}{h} \quad \text{and} \quad D_{ij}^{+x}T = \frac{T_{i+1,j} - T_{i,j}}{h} \quad (2.1.4)$$

For the calculation of each arrival time  $T_{ij}$  the following method is used [50]. With  $a = min(T_{i-1,j}, T_{i+1,j})$  and  $b = min(T_{i,j-1}, T_{i,j+1})$ , the values of  $T_{ij}$  are calculated by applying equations 2.1.5 and 2.1.6.

$$\text{If } \frac{1}{F_{ij}} > |a - b|, \quad \text{then} \quad T_{ij} = \frac{a + b + \sqrt{2(\frac{1}{F_{ij}})^2 - (a - b)^2}}{2} \quad (2.1.5)$$

$$\text{else} \quad T_{ij} = \frac{1}{F_{ij}} + min(a, b) \quad (2.1.6)$$

This way each arrival time can be calculated iteratively. The algorithm will be explained further in the following section.

## 2.2 FMM algorithm

In this section the algorithm will be delineated at first and then a comprehensive description will be given on how it was implemented.

### 2.2.1 Algorithm outline

The central idea behind the fast marching method is to systematically construct the solution in a downwind fashion, the key is the observation that the upwind difference structure of equations 2.1.5 and 2.1.6 means that the information propagates one way, that is, from smaller values of  $T_{ij}$  to larger values. The algorithm is made fast by confining the building zone to a narrow band around the front. The idea is to sweep the front ahead in an upwind fashion by considering a set of points in a narrow band, freezing the values of the existing points and bringing new ones into the narrow band structure. An important part is the selection of which grid point in the narrow band to update. [51]

During the construction of the solution each grid point can be assigned three possible states:

- Known : the time value is defined and cannot be changed, this set of points constitutes the front and is considered the upwind side. To begin the algorithm only one initial point is known and more are added iteratively until each point of the grid is known.
- Near : these points constitute the narrow band, they are neighbors of known points and contain computed trial values of times.
- Far away : these points are outside of the narrow band and constitute the downwind side.

Starting from an initial known value, that will be referred to as seed point, the algorithm marches downwind from this point by adding its 4-connected neighbors to the narrow band and computing their time values. To continue marching downwind a point in the narrow band must be chosen and set to known, consequently fixing its time value and proceeding to adding its 4-connected neighbors to the narrow band. This way the front can be expanded until all points are known, what remains is the criteria by which the new known point is chosen within the narrow band. The answer lies in the observation that the smallest time value in the narrow band must be correct, this algorithm works because the process of recomputing the time values at upwind neighboring points cannot yield a value smaller than any of the known points. Thus the solution marches outward, always selecting the narrow band grid point with minimum trial value for  $T_{ij}$  and readjusting the neighbors.

The update procedure is shown in Figure 2.2, where the black points correspond to known values, the grey ones correspond to points in the narrow band and the white ones are far away.

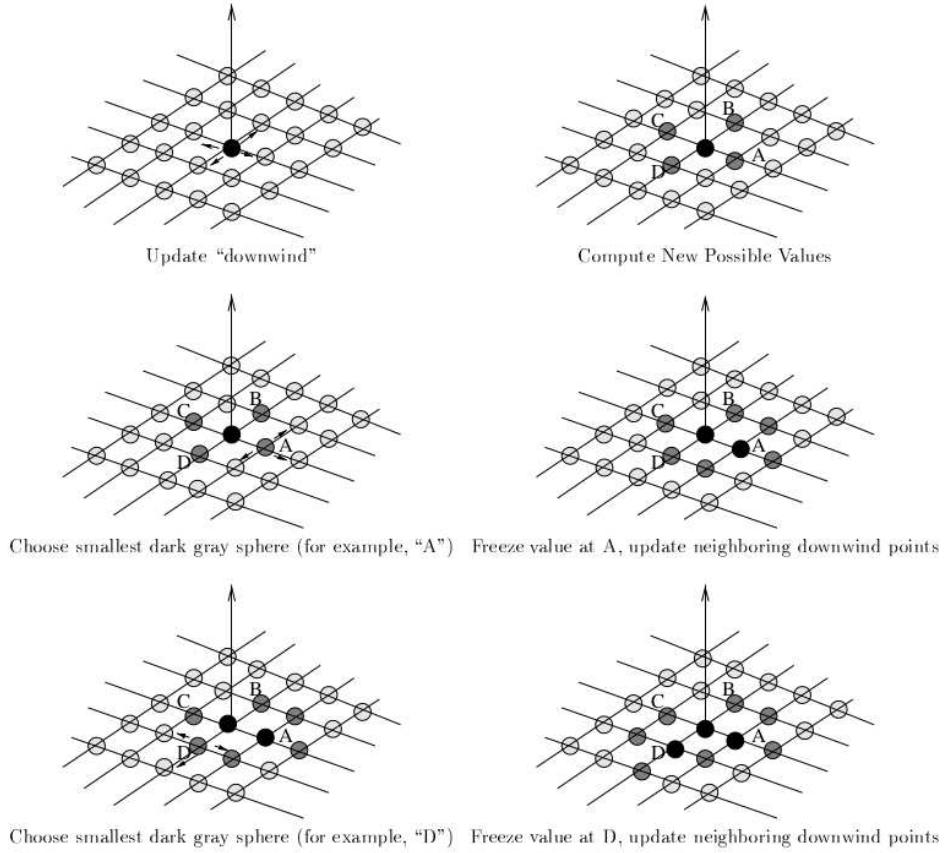


Figure 2.2: Update procedure for fast marching method [51]

The technique is easiest explained algorithmically [27] . Take a grid of equidistant points with unitary distance and with  $F_{ij}$  being the value of the speed function defined for each point in the grid. At first all the points are set to their initial value and then the algorithm marches forward in a loop.

- Initialisation

1. (Known points): firstly a set of seed points is defined, which will be the initial front, the most simple case is just one seed point, for multiple seeds the procedure is the same. Let  $A$  be the set of seed points, set  $T_{ij} = 0$  for  $(i,j) \in A$

2. (Near points, narrow band): let narrow band be the set of all grid points that are neighboring the known points, in the simplest case its just the 4 neighbors of the single seed point. Set  $T_{ij} = \frac{1}{F_{ij}}$  for all points in the narrow band, which are the first calculated arrival times.
3. (Far away points): let far away points be defined as all the points in the grid that aren't known or near. Set  $T_{ij} = \infty$  for all far away points.

- Marching Forward

1. Begin loop: let  $(i_{min}, j_{min})$  be the point in the Narrow Band with the smallest value of T.
2. Add the point  $(i_{min}, j_{min})$  to the set of known, remove it from narrow band.
3. Tag as "new neighbors" any points  $(i_{min}-1, j_{min}), (i_{min}+1, j_{min}), (i_{min}, j_{min}-1), (i_{min}, j_{min}+1)$  that are either in narrow band or far away. If the neighbor is in far away, remove it from that set and add it to the set narrow band.
4. Recompute the values of T at all the "new neighbors" according to equations 2.1.5 and 2.1.6
5. If all points in the grid are known, then end the loop, otherwise return to the top of the loop.

The key to rendering this technique efficient lies in locating the point in the narrow band with the smallest value for T in a fast way. This is done via a minimum heap structure, which is a binary tree for which every parent node has a value less than or equal to any of its children. The interesting property of a min-heap is that its smallest element is always the root. The values of T for each point in the narrow band are kept in a min-heap, and the smallest value is easily obtained by popping its root element.

Let's imagine a grid where the FMM algorithm is applied starting from multiple seed points. For each seed a region is defined, which at first only contains the seed and then expands to all the points that the seed reaches more quickly than the other seeds. It is as if the seeds are competing for the points in the grid, the quickest one to reach the point can add it to its region. This functionality can be added to the algorithm by taking note whenever a new time is calculated, of the seed which it propagated from. When the algorithm is finished and all points are known, every point will be assigned to a region and a region map is created of the same size of the grid, which creates sort of a "stained glass" effect.

In Figure 2.3 an example is shown, in this case there are two seeds A and B that propagate using the fast marching method, the darker colors symbolize the known points and the brighter colors the narrow band, the arrows show which points in the narrow band have been just updated. In (a) the speed function is shown, the values of  $F_{ij}$  are higher on the right side, which means that propagation will be quicker on that side. Seed A is assigned and its neighbors times are calculated and added to the narrow band (b), at the next iteration the same thing happens with seed B and the contested point in the middle gets added to region B because the calculated time to reach it from seed B is lower. At the next 4 iterations (d,e) the neighboring points of seed B get set to known and added to region B because their arrival times are lowest (due to the fact that the speed function has higher values on that side). The next 3 iterations the neighbors of seed A get set to known and added to region A, except for the point on the right side which is already known and belongs to region B. The next 2 iterations another two points that are between region A and B get set to known and added to region B (h). The remaining points are set to known and added to the correct region (i), in the end all the points are known and depending on which seed had the shortest arrival time they belong to region A or B.

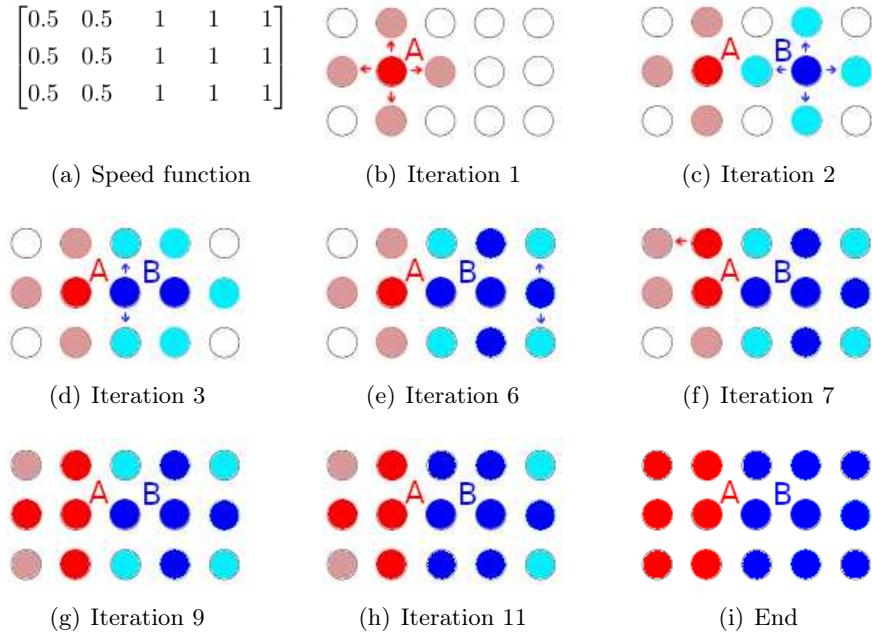


Figure 2.3: Example of region assigning

### 2.2.2 Algorithm implementation

Although there are existing implementations of the fast marching method, for this work the algorithm was implemented from square one. The reason behind this choice was to have more flexibility and also to be able to add the region functionality, in addition this approach helps to understand better the inner works of an algorithm.

The speed matrix  $F$  and a set of seed point coordinates will be the inputs, the algorithm is run iteratively by using the techniques previously discussed, until all points are known. Finally the end result will be two matrices with the same size as  $F$ :

- $T$  : the times matrix containing all the shortest arrival times.
- $R$  : the regions matrix carrying the information of which point belongs to which region.

The implementation is object based, and each class will be discussed with its functions and data structures in the following sections.

#### 2.2.2.1 Status

This class is made up of a matrix the size of  $F$  called the status matrix, it keeps track of the points in the known, near and far away sets. Each element in this matrix can have three possible values:

- -1 : means that the point belongs to the far away set
- 0 : means that the point belongs to the near set
- 1 : means that the point belongs to the known set

The status matrix is constructed with all -1 (far away) values, and the elements corresponding to seed points are set to 0 (near). Besides the matrix there is also a list called new neighbors list that is reset each iteration, and keeps track of the points in the near set with newly calculated time values. There are also a couple of functions that help manage the status matrix and new neighbors list:

- *setknown* : sets a near point to known by changing its value in the status matrix from 0 to 1.
- *setnear* : firstly the function resets the new neighbors list, then takes the adjacent points that have status different than 1 of the newly added known point to 0, and also adds them to the new neighbors list.

These two functions are always called one after another, each time a new known point is added its 4-connected neighborhood is added to the near set and to the new neighbors list, of course if the points aren't already known. Being  $N$  the number of seed points, in the first  $N$  iterations the seed points are added to known ( because their arrival times are zero, so the smallest ) and all their neighbors are added to near. The first  $N$  iterations conclude the initialisation of the grid (all seeds are known, and their neighbors are in the narrow band) and at the next iteration a new point with the shortest calculated travel time will be added to known.

An example of how the status matrix looks during the execution of the algorithm is shown in Figure 2.4, in this example the seeds are two,  $N = 2$ . At first the matrix is constructed with all -1 values (a), afterwards the two seed points are set to near (b), after two iterations both seeds are known and their neighbors are near (c). The loop continues (d) until all points are known (e).

$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$
(a) Construction	(b) Start
$\begin{bmatrix} -1 & 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & -1 & -1 \\ -1 & 0 & -1 & 0 & -1 \\ -1 & -1 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 & 0 \end{bmatrix}$
(c) After 2 iterations	(d) After 7 iterations
$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	
(e) End	

Figure 2.4: Example of status matrix with two seeds

### 2.2.2.2 Times

This class is also made up of a matrix with the same size as  $F$ . It keeps track of the arrival time values for every point. When it is constructed the values of the seed points in this matrix are set to 0, and the rest to infinity. There are also a couple of functions that manage the times matrix:

- *addtime* : this function adds a calculated arrival time value  $T_{ij}$  to the times matrix at the given point
- *calculatetime* : this function calculates the arrival time  $T_{ij}$  at the given point using equations 2.1.5 and 2.1.6

In Figure 2.5 is an example on how the times matrix evolves over time. Initially only the seeds have a finite time value of zero and the rest of the points are set with a value of infinite. The front propagates and the times increase the further they are from the seeds (b,c), until the end is reached (d) and the times matrix is complete.

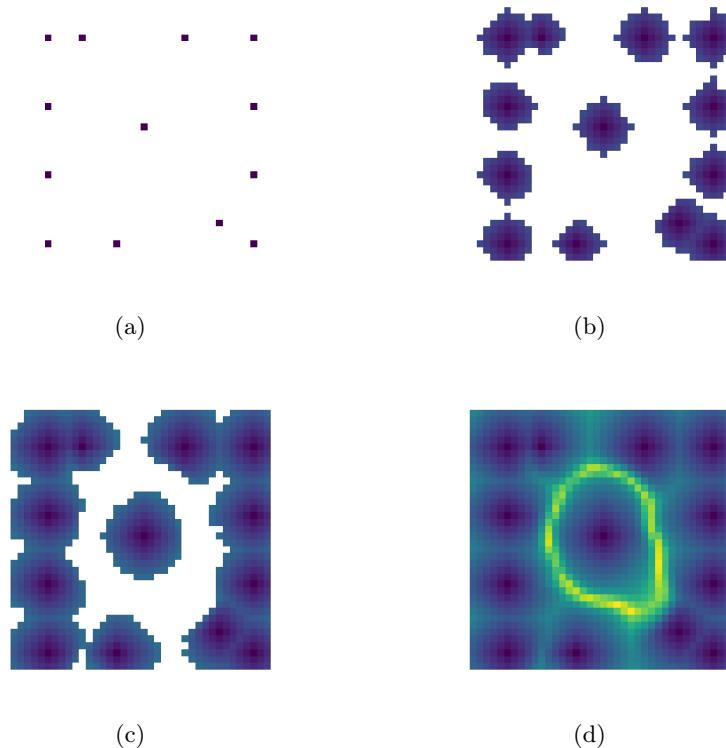


Figure 2.5: Example of times matrix evolution

### 2.2.2.3 Heap and Region

This class consists of two objects. One of the objects is a heap, that contains and organises the calculated arrival times  $T_{ij}$  in the near set (narrow band). The heap is represented as a Python list and is managed using functions from the *heapq* Python module, which provides an implementation of the heap queue algorithm. Besides the calculated arrival time value of each point in the near set, it also contains the coordinates of each near set point and an integer which identifies which seed the time values propagated from (called seed id). Actually each element of the heap is a tuple with four items, but only the arrival time is used for ordering and maintaining the heap structure. The heap is constructed with  $N$  elements,  $N$  being the number of seeds, these first elements all have arrival time zero and correspond to the seeds. Other than making the minimum arrival time value search efficient the heap also keeps a record of which arrival times originate from which seeds. The element at index zero of the heap is always the one with the minimum arrival time, each iteration the point corresponding to this element is added to the known set and the seed id is added to the regions matrix.

The other object of the class is this regions matrix, which is the same size as  $F$ . This matrix keeps track of the regions and is constructed as all -1 values. Whenever a new arrival time is added, the seed id is copied to the regions matrix in the correct position, at the end this matrix contains all positive integer values that label each region.

Several functions are defined in this class that help manage the heap and regions matrix:

- *push*: with this function, new elements can be added to the heap while retaining the correct structure. It also keeps up the regions matrix by copying the seed id of the newly added value at the coordinate that corresponds to the point.
- *replace*: this function deletes a specific value from the heap and then calls *push* to replace it with a new value.
- *check*: the purpose of this function is to check whether there already exists a calculated arrival time value for the near point, this is done by searching through the heap. If no such value is found, then *push* is called to add the new time. Otherwise the found value is compared to the newly calculated value, if the new time is lower than the old value *replace* is called to adjust the heap accordingly.

In Figure 2.6 is an example on how the regions matrix evolves over time. Starting from the seeds, the regions expand until the matrix is full.

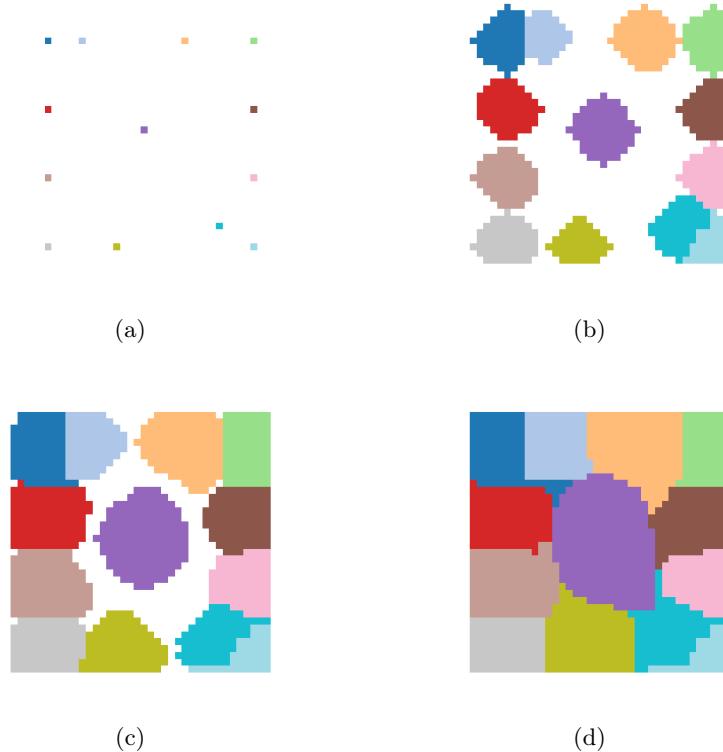


Figure 2.6: Example of regions matrix evolution

#### 2.2.2.4 Implementation

Putting all three classes together the fast marching method can be executed. Firstly the status, times and regions matrices are constructed, and the proper values are set for the seeds:

- Status : constructed with all -1 values, seeds are set to 0.
- Times : constructed with all  $\infty$  values, seeds are set to 0.
- Regions : constructed with all -1 values, seeds are set to their id, which are integers from 1 to  $N$ , with  $N$  being the number of seeds.

Additionally the new neighbors list and heap are constructed in this manner:

- New neighbors list : constructed as an empty list
- Heap : constructed with  $N$  elements corresponding to the seeds.

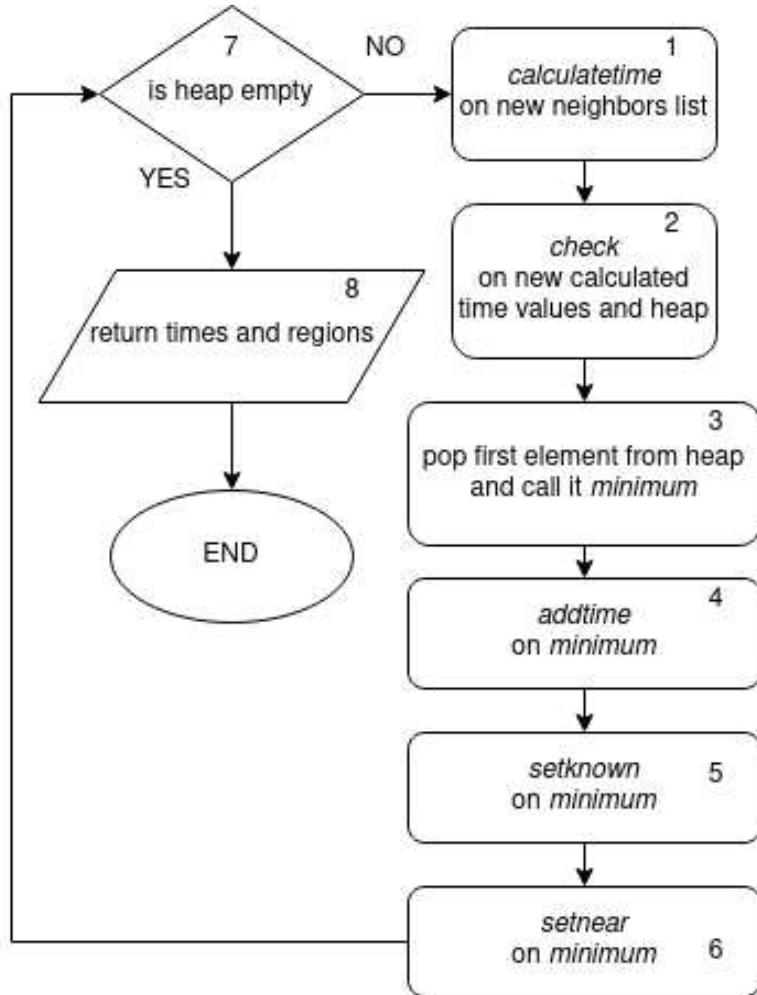


Figure 2.7: Loop flowchart

Afterwards the loop depicted as a flowchart in Figure 2.7 proceeds as follows:

1. For each point in the new neighbors list, a new arrival time is calculated using *calculatetime*
2. The newly calculated arrival times are checked against existing ones in the heap using *check*, and are added if new or if there are existing values they are confronted and the smallest is kept.
3. The first element from the heap is popped, and it is a tuple containing time value, point coordinates and seed id. This tuple will be called *minimum*
4. The arrival time from *minimum* is added to the times matrix using *addtime*

5. The point corresponding to the coordinates in *minimum* is set to known using *setknown*
6. Its neighbors that are not known are set to near using *setnear*, and are also used to update the new neighbors list
7. If the heap is empty, this means that there are no more trial values, and all points are known, meaning that the algorithm has finished. Otherwise the loop continues.
8. The results are the finished times and regions matrices.

## 2.3 Segmentation - preparation

Before applying the fast marching method two things are necessary, a speed function and a seed grid. The speed function depends on the gradient of the input, and the seeds are distributed on a grid based on the local gradient mean.

### 2.3.1 Speed function

Since the speed function is an image with the same dimensions as the input, it can also be referred to as speed image. A good propagation speed image for segmentation is close to zero near object boundaries and relatively high in between. In order to get a good speed image a number of tools from the SimpleITK Python library [52] were employed.

At first the input image is denoised with an anisotropic diffusion filter, which removes noise without influencing edges, for this purpose the function *sitk.CurvatureAnisotropicDiffusionImageFilter* was used.

To compute magnitude of the image gradient at each pixel location the function *sitk.GradientMagnitudeRecursiveGaussianImageFilter* was applied, which is equivalent to first smoothing the image by convolving it with a Gaussian kernel (with  $\sigma = 1$ ) and then applying a differential operator. Internally this is done by applying an IIR( Infinite Impulse Response) filter that approximates a convolution with the derivative of the Gaussian kernel. The speed image needs to be an inverse function of the image gradient, the function chosen was an exponential with negative exponent, specifically  $F(x, y) = e^{-2\nabla(x, y)}$  (the value -2 was chosen empirically).

After normalising the speed image to a range from zero to one, the values are near zero at the boundaries and in homogeneous regions they are close to one. An example is shown in Figure 2.8.

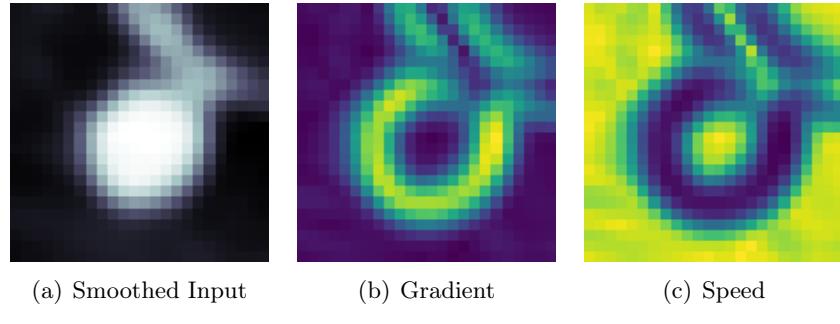


Figure 2.8: Example of speed image

In case of juxtapleural nodules an issue arises, since the lung wall was removed during the lung segmentation step in the preprocessing section, there will be a high image gradient because the image has a sharp transition, and this will influence the segmentation negatively. To render the image gradient, and subsequently speed image, invariant to this undesired effect a padding scheme was applied to the input before calculating the gradient.

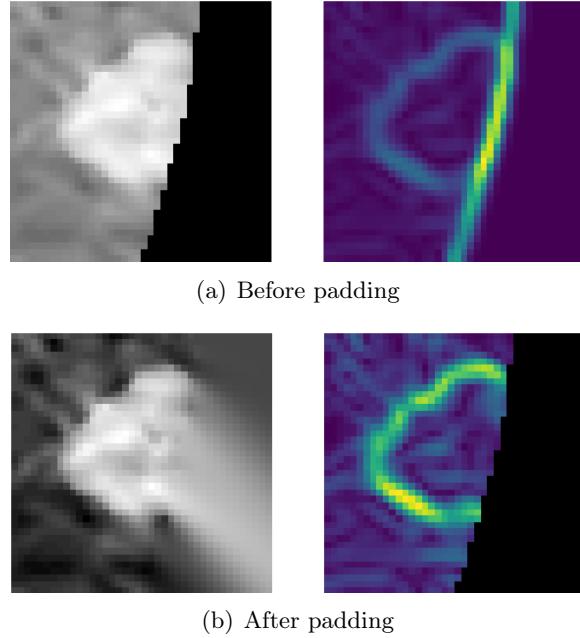


Figure 2.9: Effects of padding on image gradient

In Figure 2.9 is an example which shows the effect that this padding process has on the image gradient, before padding (a) the image gradient is a lot more accentuated on the lung wall than on the rest of the boundary, since the information about how the lung wall is segmented is already available

this only interferes with the lung segmentation. By applying the padding scheme the gradient along the nodule boundary becomes more accentuated, and along the lung wall it becomes negligible, in (b) the padded input image is shown along with the gradient obtained using the padded input and after applying the mask. The padding is anisotropic and gives larger intensity values along the diagonal, but this undesired effect is negligible since the only relevant part of the padding is along the lung wall, and the rest gets cropped out after the gradient is calculated. The padding scheme was implemented in a simple way: starting from the zero value pixel (the pixels with zero intensity correspond to the outside of the lung) closest to the upper left corner of the image, the new pixel value is calculated as mean of all non zero 4-connected neighbors. This continues iteratively by going over each zero value pixel, resulting in a padded image that won't have a high gradient along the lung wall, effectively ignoring this border.

To make sure that the segmentation algorithm does not confuse this padding with tissue from a regular nodule, a binary mask is saved that takes this into account, and is applied at different stages to make sure that the region of the image that does not contain lung tissue will be ignored by the segmentation algorithm. Going forward, juxtapleural nodules won't need any more special consideration when it comes to segmentation.

### 2.3.2 Seed grid

Now that the speed image is ready, the seeds need to be chosen from which the front will propagate using the fast marching method. An optimal seed grid should have a higher concentration of seeds in uniform areas in the image, a good way to satisfy this condition is to use the image gradient as a criterium for choosing seed point, the image gradient is normalised to a range from zero to one. Another thing to consider is that the images that need to be segmented are low resolution, this means that in order to obtain accurate results a dense seed grid is needed. Keeping these two requirements in mind, the algorithm works in the following way:

- At first the mean of the image gradient is calculated and will be used as a threshold moving and deleting seed points, this variable will be called *total grad mean*.
- The seed grid is initialised with equidistant points, the distance can be varied, but to get the densest possible grid the minimum value was chosen which equals to three. For each seed a local area is assigned, which corresponds with the points that are closest to it, with distance equal to three the local area is the 8-connected neighborhood of the seed. For each seed's local area a mean is calculated using the gradient values within that area, these variables will be called *local grad mean( $i$ )*, with  $i$  being an integer that identifies the seeds.

- Based on the value of  $local\ grad\ mean(i)$  for seed  $i$ , three possible things can happen:
  1. If  $local\ grad\ mean(i) > \beta \cdot total\ grad\ mean$  then seed  $i$  is deleted from the grid. The reasoning behind this is that if the local area of the seed has a relatively high gradient mean, this suggests that the local area is not sufficiently uniform and the seed needs to be removed.
  2. If  $\alpha \cdot local\ grad\ mean(i) < total\ grad\ mean$ , the seed is kept as is, the local area is sufficiently uniform.
  3. If  $\alpha \cdot total\ grad\ mean \leq local\ grad\ mean(i) \leq \beta \cdot total\ grad\ mean$ , the seed point is shifted. Instead of removing these points a better alternative was to slightly shift them, resulting in a denser grid. Shifting is done in the following manner: for each point in the local area the 8-connected neighborhood gradient mean is calculated, the one with the minimum value is chosen as the new seed. This results in the seed being moved towards a more uniform area of the image.

The parameters that define the thresholds for keeping/shifting/deleting seeds have been fixed to the values of  $\alpha = 1$  and  $\beta = 2$ . These values have been selected after extensive experimentation.

An example is shown in Figure 2.10, since the nodule is juxtapleural the outside of the lungs is ignored in the grid creation, and the addition of shifting renders the grid denser and better distributed.

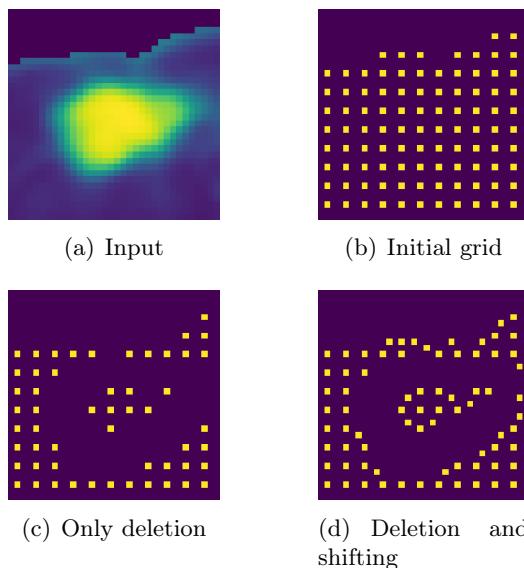


Figure 2.10: Seed grid generation example

## 2.4 Segmentation - region merging

After applying the fast marching method a region map is obtained with the same number of regions as there are seeds. To get a more detailed segmentation it is necessary to have a seed grid that is dense, meaning that there will be a large number of small regions. The segmentation mask is obtained via a region growing method [53]. Starting from a region that is sure to belong to the nodule, neighboring regions are added recursively if they match a certain criteria. The selected criteria is the mean intensity of each region, if the mean intensity difference between two regions is higher than the threshold this means that there should be a boundary between them, conversely if it is lower than the threshold they should both belong to the nodule. Since the regions are relatively small, neighboring regions will often have a mean difference that is lower than the threshold even in presence of a boundary. In many cases this creates a problem, a threshold that is too low will not include all the regions containing the nodule and a threshold that is too high will cause the merging to bleed through the boundary ultimately merging all the regions of image. This is shown in Figure 2.11, where the regions are displayed in different colors, in (b) not enough are merged to provide a good segmentation and in (c) the merging goes over the boundary and takes the background as well.

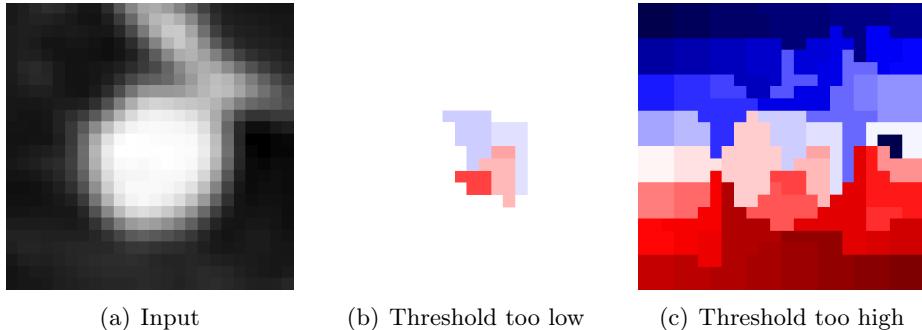


Figure 2.11: Region merging errors

A way to remedy the aforementioned issue, without sacrificing seed density, would be to group together regions into larger ones. Looking at the way the seed grid is constructed, seeds in uniform areas are packed closer together and should be grouped together, Euclidean distance will be used to decide which regions go together. The most obvious choice is using K-means to put regions together into clusters.

### 2.4.1 K-means clustering

K-means clustering [54] is the most classical partition-based clustering method. The basic idea of K-means is to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean. Iteratively, the values of centroid of clusters are updated one by one until the best clustering results are obtained. The K-means algorithm takes Euclidean distance as the similarity measure and the clustering criterion function is within cluster sum of squares. The algorithm divides a set of  $N$  samples  $X$  into  $K$  disjoint clusters  $C$ , each described by the mean  $\mu_j$  of the samples in the cluster. The means are commonly called the cluster centroids, the K-means algorithm aims to choose centroids that minimise the clustering criterion expressed in equation 2.4.1:

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|)^2 \quad (2.4.1)$$

So, starting from a set of centroids  $\mu_j$ , the distances of all samples  $X$  to each centroid are computed, and each sample is assigned to its closest centroid, a new set of centroids is calculated and this procedure is repeated until the centroids do not change anymore. Or in a more algorithmical manner:

1. Randomly select  $K$  cluster centroids  $\mu_j$
2. Calculate the distance between each sample  $x$  and cluster centers  $\mu_j$
3. Assign each sample to the closest cluster center, thus minimising the clustering criterion in equation 2.4.1
4. With  $c_j$  being the number of samples in the  $j^{th}$  cluster, recalculate the new cluster centroids using equation 2.4.2:

$$\mu_j = \frac{1}{c_j} \sum_{i=1}^{c_j} x_i \quad (2.4.2)$$

5. Recalculate the distance between each sample and newly obtained cluster centers
6. If no sample was reassigned then stop, otherwise repeat from step 3.

The default choice for initial centroids for the K-means algorithm is random, but this has the disadvantage of giving non-repeatable and varied results, often obtaining clustering that worsens segmentation performance. It makes sense to apply the same method used for the seed grid generation, because optimally the cluster centroids should stay away from the nodule border. For seed generation the distance between each point is fixed to the smallest

possible value to get a more detailed segmentation, conversely for picking the initial centroids it should be larger because several regions are bundled into each cluster. The distance can be varied to improve segmentation quality and this tunable parameter will be called "cluster density".

The K-means clustering is computed with the `sklearn.cluster.KMeans` module from the scikit-learn Python library [29], giving in input the set of initial cluster centroids K and seed points N, the results are the final cluster centers and cluster index for each seed. The regions corresponding to these seeds are merged together into larger regions using the cluster index, to avoid confusion with the previous smaller regions generated by the FMM algorithm they will be called clusters from now on.

This is demonstrated in the example in Figure 2.12, the regions (a) are transformed into clusters (b), in (c) the seeds are displayed over the input image with different colors based on the cluster they belong to.

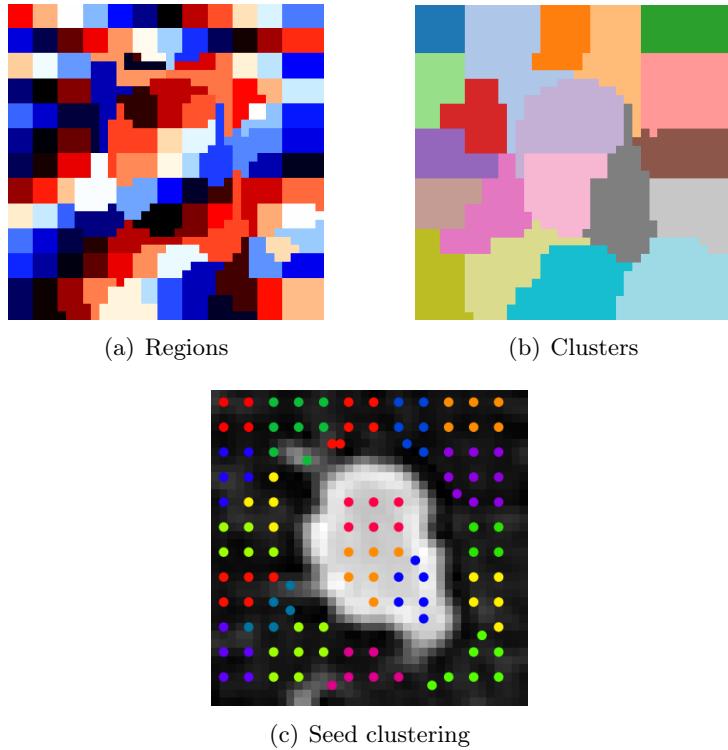


Figure 2.12: Grouping regions into clusters

Now there are clusters instead of regions, which are larger and are grouped so that more uniform regions are closer together (because the distance depends on the gradient mean). The difference in mean intensity between clusters is more pronounced, consequently it can be used as a criteria for merging the clusters and finally segmenting the lung nodule.

### 2.4.2 Merging

Similarly to the FMM implementation, the merging algorithm is also object based. The class which helps implement this method is called *Cluster*, each *Cluster* object is characterised by:

- centroid : the coordinates of the cluster centroid
- integer identifier : number ranging from 1 to K, identifying the specific cluster
- list of neighbors : provides a list of neighboring clusters
- mask : binary image where the pixels associated to the cluster are one and the rest are zero.

There are several functions which are used to manage this class:

- *add neighbor clusters*: This function populates the list of neighbors for each cluster. This is done by checking if the dilated mask of the current cluster overlaps with any of the masks of all the other clusters. To test whether they overlap, the current cluster dilated mask and possible neighboring cluster mask are multiplied and if the result is not a matrix of zeros, this means they are neighbors.
- *mean of cluster*: The mean intensity of the cluster is calculated by taking the mean of the input image times the cluster mask.
- *merge clusters*: This function recursively merges the clusters, the end result is a list of connected clusters which by union of their masks create a final segmentation mask. This list initially contains only the starting cluster. Each time the function is called on a *Cluster* object, it calculates the mean intensity difference with all its neighbors using *mean of cluster*. If the mean intensity difference between the cluster and its neighbor is smaller than a threshold (which will be a tunable parameter called "mean threshold"), then the neighbor cluster is added to the connected clusters list and *merge clusters* is called on it. This continues recursively until there are no more suitable clusters to add.

Taking the results from K-means, the *Cluster* objects are constructed, and for each object the *add neighbors cluster* function is called. Afterwards a list of connected clusters is created, initially containing only the starting cluster. Subsequently, the *merge clusters* function is called on the starting cluster and continues recursively, when it finishes it returns the list of connected clusters which is used to create a final segmentation mask.

In order to start the merging, a starting cluster needs to be chosen. When there are no obstructions or other tissue with similar radiodensity, the nodule is usually identified by a higher intensity in the image, but this is not always

the case, so choosing the cluster with highest mean intensity could fail in cases where it doesn't belong to the nodule. The images that are used as input are centered on the nodules, but in edge cases like cavitary nodules the cluster containing the central pixel is not part of the nodule. Both choices of starting cluster have issues, but they can be resolved by combining both methods. In order to choose the starting cluster, for each cluster a ratio is calculated between the mean intensity and distance of centroid to the central pixel, the cluster with the highest ratio wins and is selected as the starting point for merging. This way both closeness to the center of the image and high mean intensity are taken into account.

In Figure 2.13 an instance of cluster merging is shown, commencing from the starting cluster the algorithm adds clusters one by one until there are no more neighboring clusters that satisfy the merging criteria, finally obtaining the segmented nodule (f).

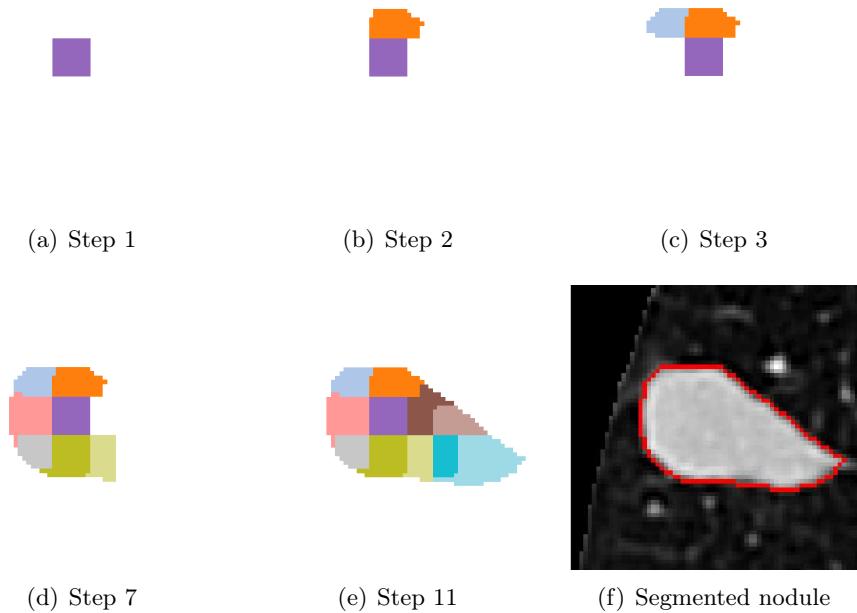


Figure 2.13: Merging of clusters

With this the segmentation algorithm is completed, and can be used to segment lung nodules.

### 2.4.3 Parameter tuning

There are several parameters which have been fixed and altering them doesn't change segmentation accuracy in a significant way. All the value were selected empirically.

- $\sigma = 1$  : before calculating the gradient magnitude the image is convolved with a Gaussian kernel with this standard deviation value.
- $\tau = -2$  : the speed function is an exponential of the gradient magnitude, the exponent  $\tau$  is negative because it needs to be an inverse function of the gradient magnitude.
- $\alpha = 1$  and  $\beta = 2$  : when generating the seed grid, these parameters determine the thresholds at which seed points will be kept, removed or shifted.
- *seed distance = 3* : this parameter determines how dense the seed grid is, the smallest possible value is picked because it provides the most detailed segmentation results. The side effects of choosing this value are mediated by the addition of K-means clustering.

Furthermore, there are two tunable parameters that can be altered to improve segmentation:

- *Cluster density* : determines how dense the cluster centroid grid is, thus determining the size and total number of clusters. Lowering this parameter can improve segmentation accuracy, but the tradeoff is that the mean difference between neighboring clusters is less pronounced and thus an unwanted cluster ( does not contain nodule tissue ) could be merged. The default value of this parameter is 7.
- *Mean threshold* : determines which clusters are suitable for merging and will be part of the segmentation mask. This parameters gives the criteria for merging clusters, if the mean intensity difference between a cluster and its neighbor are below the threshold they are merged. A suitable threshold is needed for good segmentation performance, the default value of this parameter is 0.15

To make parameter tuning more handy, two trackbars were added to the segmentation program. The trackbars are initially set to the default values and have ranges from 0.00 to 0.30 for *mean threshold* and from 3 to 10 for *cluster density*. The trackbars together with a preview of the segmentation result are shown in Figure 2.14.

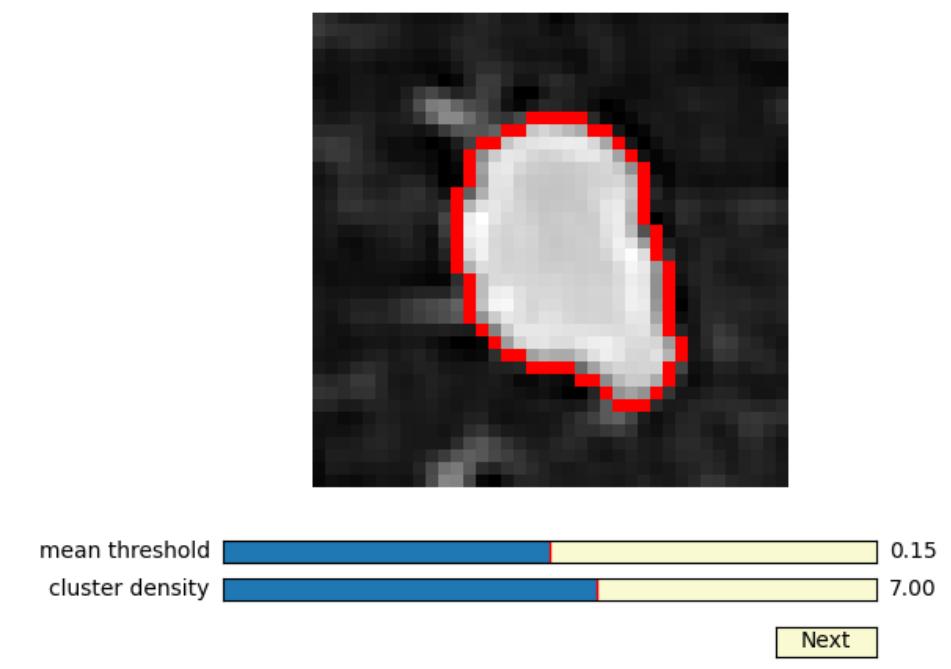


Figure 2.14: Trackbars for parameter tuning

# Chapter 3

## Evaluation

After applying the preprocessing method to prepare the data, and implementing the fast marching method to segment the nodules, finally the results can be evaluated. There will be two separate evaluation procedures for the two datasets:

- Subjective evaluation : performed using data obtained from the public LIDC dataset and with the help of a radiologist. A questionnaire was compiled where each segmentation instance was shown and the radiologist was asked to rate each instance with three scores.
- Objective evaluation : carried out by using the phantom dataset, which was specifically generated for this study. Due to the simplicity of the phantom nodules, the images were altered to give more complexity. The results are evaluated by comparing ground truth and the segmentation mask using different metrics.

Having two separate datasets with their own evaluation method has certain advantages. By analyzing and comparing both results the segmentation algorithm can be evaluated in greater depth. Moreover, it is also useful to compare what is obtained by an evaluation performed by a radiologist on real patient CT data, and what is obtained from an automated evaluation performed on artificially generated nodules. In this fashion dissimilarities between the two methods and datasets can be examined, highlighting which features are prioritised when evaluating lung nodule segmentation accuracy.

### 3.1 Comparison method

In order to establish how well the segmentation algorithm performs, another method is necessary for comparison. With this in mind, the *activecontour* tool from Matlab [30] Image Processing Toolbox was used. This tool segments the image into foreground and background using active contours. The active contours technique, also called snakes, is an iterative image segmentation algorithm. Using the active contour algorithm, initial curves on an image need to be specified and then the curves evolve towards object boundaries. As input it requires the image to segment and binary mask specifying the initial contour, after the stated number of iterations the final contour is returned as a binary mask.

The active contour method used for segmentation is the Chan and Vese region-based energy model described in [9], the *activecontour* function uses the sparse field level set method, similar to the method described in [55], for implementing active contour evolution.

In Figure 3.1 an example of contour evolution is shown. The initial contour is a square whose sides have a distance of 5 pixels from the image border (a), *activecontours* continues to evolve the contour until 150 iterations are reached (d) and the nodule is segmented. The number of necessary iterations is variable, and has been set to 300 to make sure that the algorithm has enough time to accurately segment the nodule.

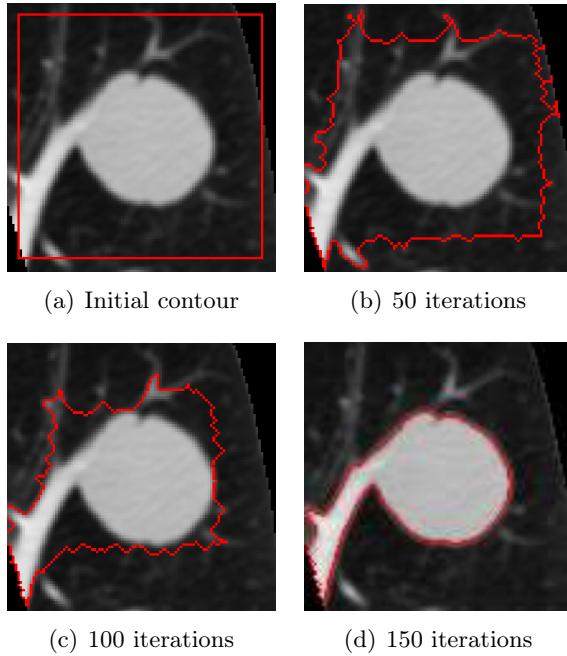


Figure 3.1: Example of *activecontours* segmentation

## 3.2 Subjective evaluation

The LIDC database also contains annotations which were collected during a two-phase annotation process using 4 experienced radiologists. Each radiologist marked lesions they identified as non-nodule, nodule  $< 3$  mm, and nodules  $\geq 3$  mm. [56]. A ground truth can be estimated using these annotations, but trying to extract a ground truth by combining four different readings adds unnecessary complexity and can introduce errors [57].

Since for this study there was a radiologist available to evaluate the results, a more direct method was chosen. Normally the radiologist would draw the ground truth of each nodule manually and then it would be compared using a metric, but manual annotation takes time and can be cumbersome, instead a different technique was devised based on MOS.

A Mean Opinion Score (MOS) is a numerical measure of the human judged overall quality of an event or experience. In telecommunications, a MOS is a ranking of the quality of voice and video sessions. Most often judged on a scale of 1 (bad) to 5 (excellent), these scores are the average of a number of other human scored individual parameters. It can be utilized to evaluate the quality of almost any visual or audio media, it has been used to evaluate the quality of medical images [58] and also to evaluate image segmentation quality [59].

### 3.2.1 Dataset construction

The LIDC dataset contains 1018 cases, which is large number of data to evaluate, thus a smaller representative subset was chosen. The nodules contained in this subset were hand picked in order to have a varied set of nodules with different sizes and characteristics. Lung nodules can be classified in a number of different ways [60], based on their appearance and clinical significance. For sorting the nodules in different categories a simple classification was devised based on visual appearance and by following the glossary of terms for chest imaging proposed by the Fleischner Society [61]. A representative subset has been compiled and sorted in the following categories:

- Round : these nodules have a fairly uniform image intensity and are characterized by a simple convex shape. They are a kind of solid nodules, the most common type, characterized by homogeneous soft tissue attenuation. Nodules with this smooth almost round shape are more likely to be benign.
- Irregular : these nodules also have a uniform intensity and belong to solid nodules, but have a more irregular shape. Nodules with irregular, lobulated or spiculated borders are associated with a progressively higher probability of malignancy than those with a smooth border.

- Semi-transparent : the pixels containing the nodule have a highly varied image intensity and are not uniform. These correspond to ground glass opacifications (GGO) and are a subset of pulmonary nodules or masses with non-uniformity and less density than solid nodules. They are usually described as either non-solid or part-solid (subsolid), and have a higher probability of malignancy.
- Cavitary : these nodules are ring shaped and identified by a cavity with darker intensity in the image. A pulmonary cavity is a gas filled area of the lung in the center of a nodule. The most commonly encountered cavitary nodules are malignant.

In Figure 3.2 an example is shown for each of the four categories.

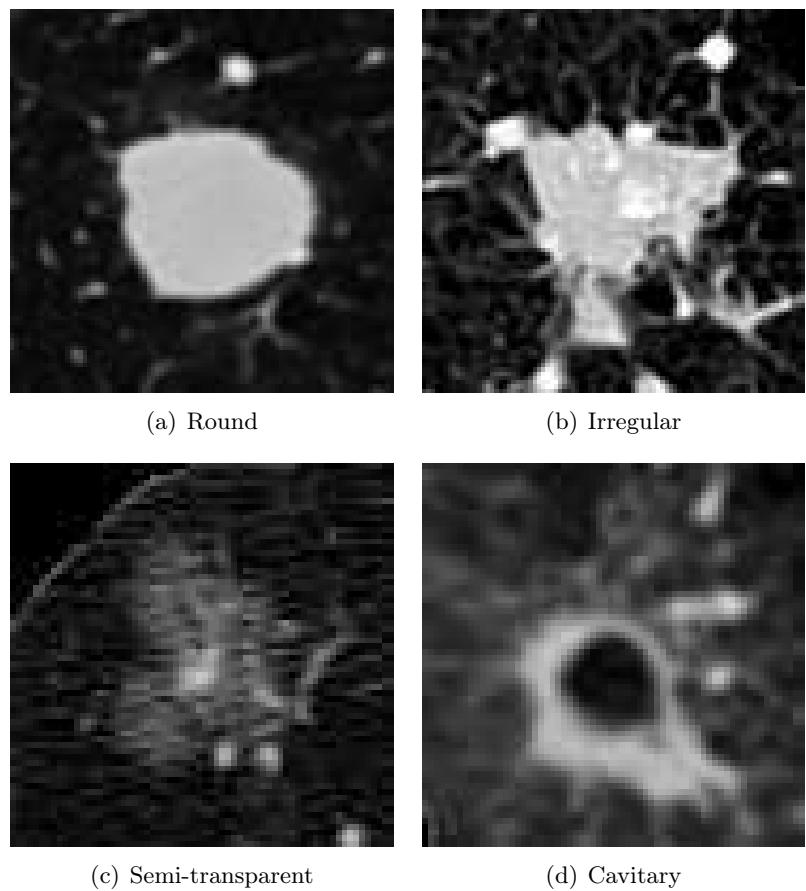


Figure 3.2: Examples of LIDC nodules from each category

### **3.2.2 Evaluation method**

A four part questionnaire (one for each category) was devised in order to evaluate the region-based fast marching method and for the preservation of diagnostic information. It was filled out by a radiologist from Tianjin Chest Hospital. These are early experiments and for now only a single subject has been involved in the subjective evaluation, for more consistent results observations from multiple subjects would be necessary.

The questionnaire had the following structure:

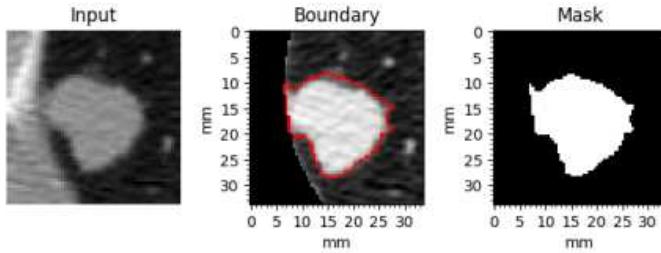
- Introduction : a paragraph that explains the purpose of the document
- Data : a paragraph that gives an overview on how the data was processed and how the nodules were segmented.
- Rating : in this section the rating scale was established. The segmentation quality was rated on a scale from 1 to 5, where 1 is the worst possible score and 5 the highest. In previous studies where MOS was used to evaluate image segmentation [58][59], a single score was used. To obtain a more insightful evaluation, for each segmentation instance three different scores were assigned. Since this kind of evaluation has never been done before for lung nodule segmentation, the scores have been devised while keeping in mind which aspects are the most important. The scores are described as follows:
  - Area preservation : this score quantifies how close the area of the segmented nodule is compared to the ground truth.
  - Shape preservation: this score indicates how similar the shape of the boundary of the segmented nodule is compared to the ground truth.
  - Overall diagnostic quality : this score rates the general quality of the segmentation, taking all the features of the nodule into account, it should quantify how well the segmentation captures all these features.

To quantify these scores and make the radiologists task clearer, for each part examples are shown on how different segmentation instances can be rated. In Figure 3.3 are a couple of examples, taken from the first questionnaire. A short justification for the rating is given, followed by the scores and a figure displaying the segmentation instance evaluated.

- Example 2 :**

There are small segmentation errors along the border, area of the nodule is very close to the ground truth, the segmentation looks fairly accurate except for finer detailed features

Area preservation	Shape preservation	Overall diagnostic quality
4	4	4



- Example 3 :**

Segmentation errors are more noticeable but shape and overall size are still close to reality. The result can be considered acceptable.

Area preservation	Shape preservation	Overall diagnostic quality
4	3	3

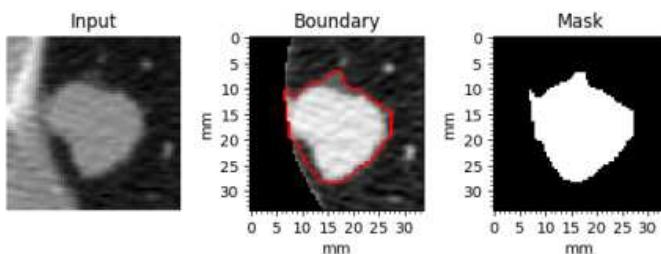


Figure 3.3: Rating scale examples from questionnaire

- Evaluation : in this final section each segmentation instance is displayed. Every nodule was segmented and shown twice, once using the fast marching method and once using active contours. The instances shown in the questionnaire have been randomly shuffled to make it impossible to know beforehand which method was used. For each segmentation instance a figure is displayed with the results, underneath it are three editable text fields to input the scores, an example is shown in Figure 3.4.

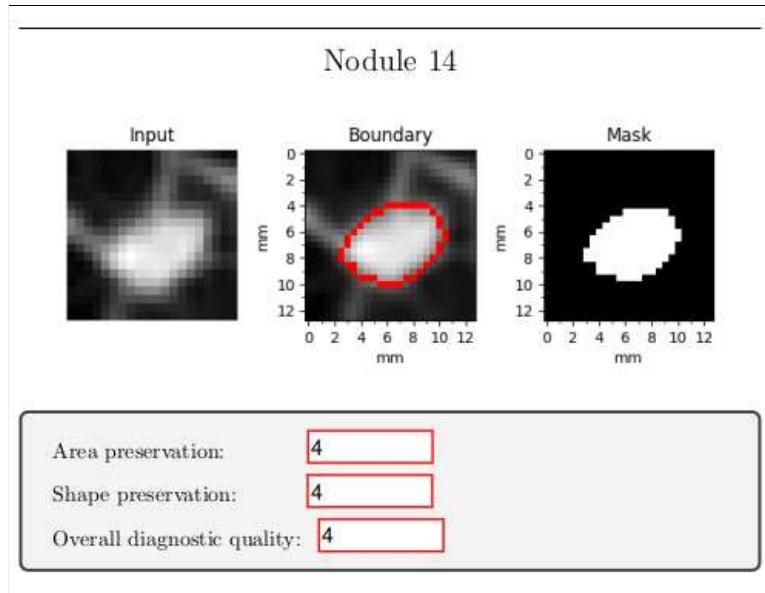


Figure 3.4: Example of segmentation instance evaluation

### 3.2.3 Results

In this section the results from the subjective evaluation questionnaire are displayed and discussed by category. For each score a graph is shown and the numerical results are recapped in a table. Following is a discussion of the results with some key examples. All the segmentation instances with the outlined borders and related evaluation scores can be viewed in appendix. The indexing of nodules is kept consistent for all graphs, tables and figures.

### 3.2.3.1 Round

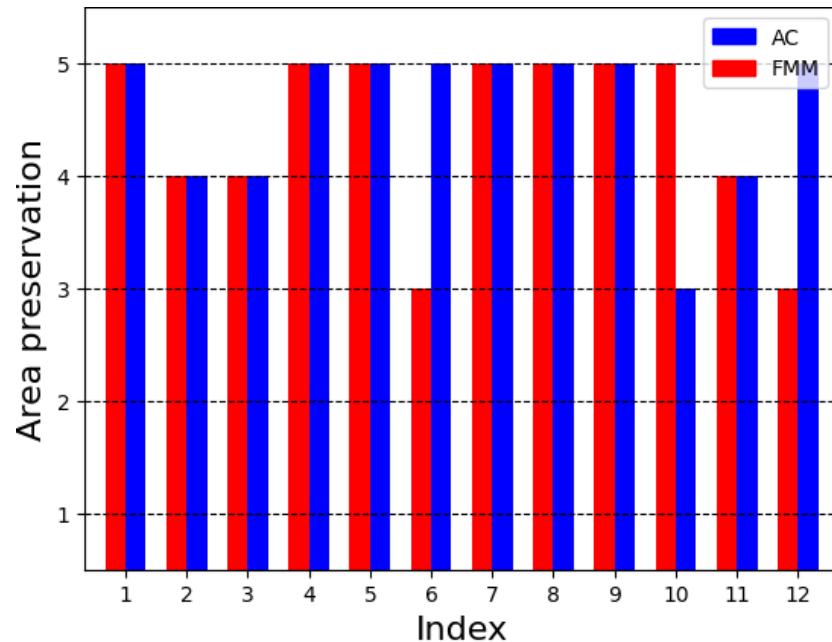


Figure 3.5: Round - Area preservation (1/2)

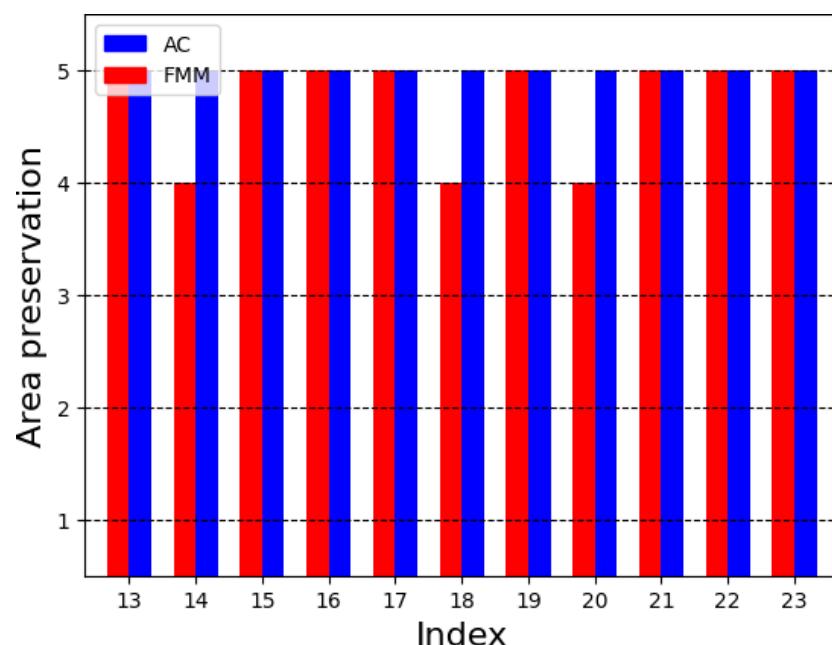


Figure 3.6: Round - Area preservation (2/2)

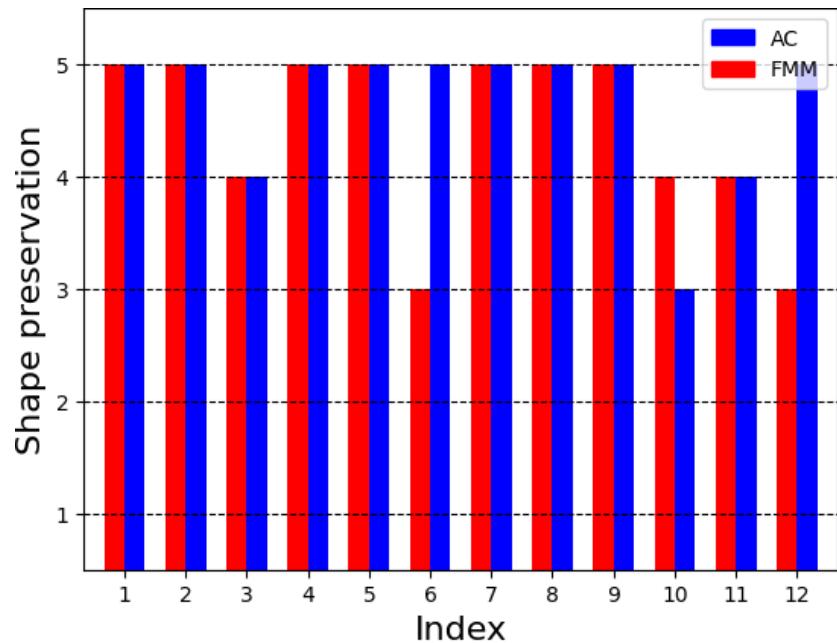


Figure 3.7: Round - Shape preservation (1/2)

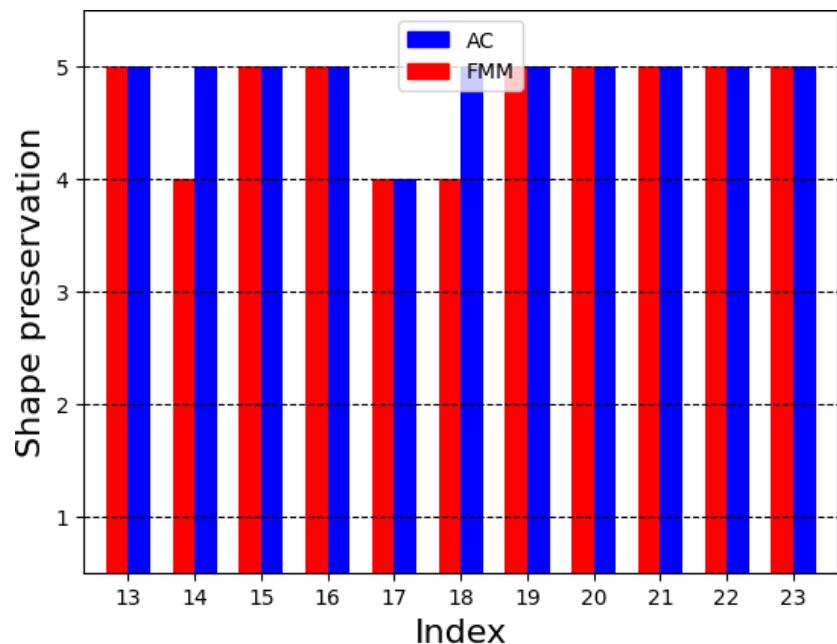


Figure 3.8: Round - Shape preservation (2/2)

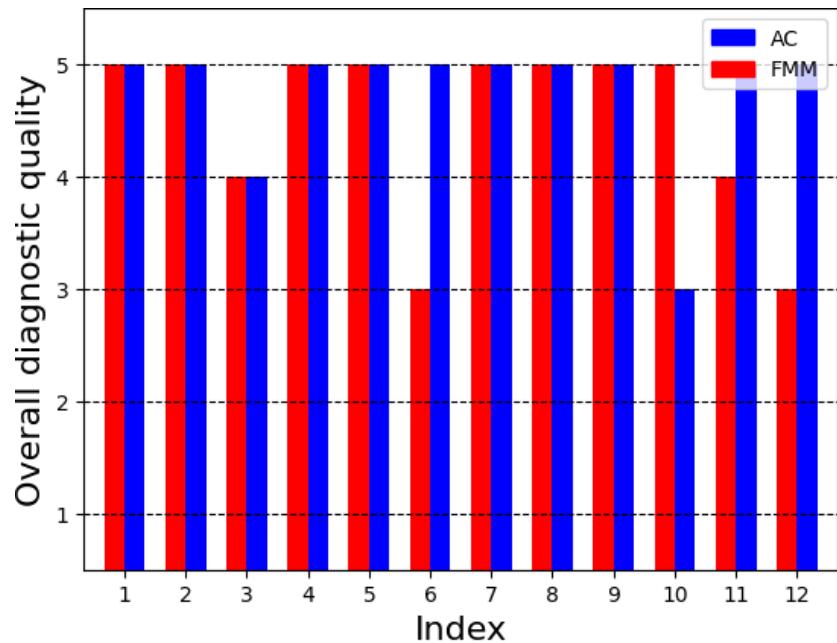


Figure 3.9: Round - Overall diagnostic quality (1/2)

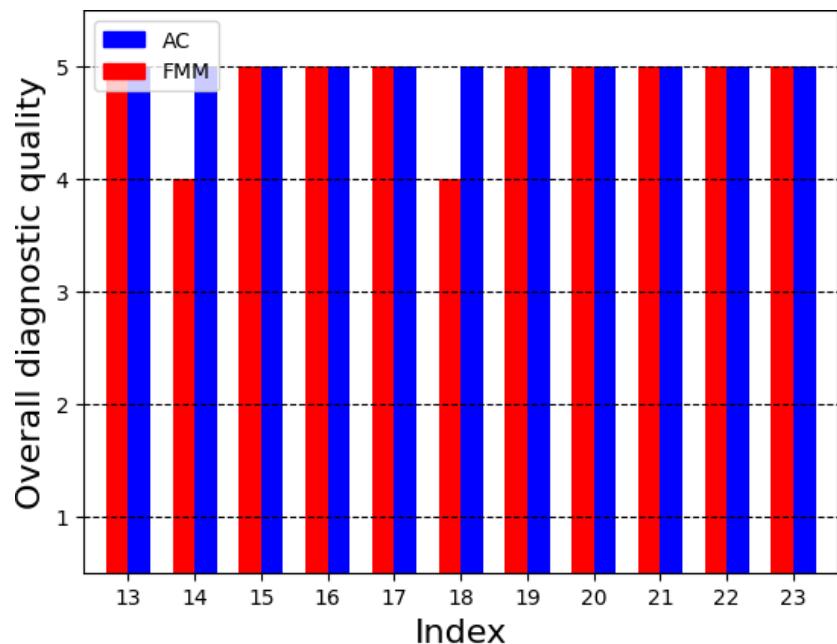


Figure 3.10: Round - Overall diagnostic quality (2/2)

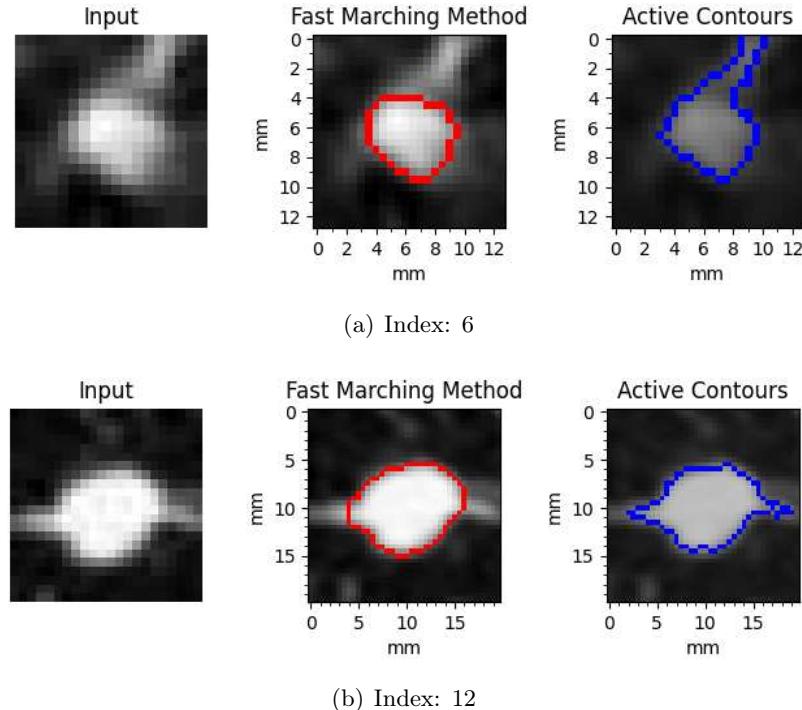
To summarize the results and make a comparison between the two methods, for each score the mean and standard deviation have been calculated and are shown in table 3.1.

	Area preservation	Shape preservation	Overall diag. qual.
FMM	$4.56 \pm 0.64$	$4.56 \pm 0.64$	$4.65 \pm 0.63$
AC	$4.78 \pm 0.50$	$4.78 \pm 0.50$	$4.86 \pm 0.44$

Table 3.1: Round subjective evaluation results summary

These round type nodules are easy to segment because of their simple shape and regular border, both segmentation algorithms perform very well and the mean average scores are close to 5 (Excellent). AC has a slightly higher mean for all three scores.

In Figure 3.11 are the instances where AC and FMM scored differently. In (a-d) FMM segmentation received a lower rating for all three scores, because it didn't encompass enough of the surrounding tissue, which the radiologist deemed important to include. In (e) FMM received a lower score in just area preservation, because it failed to cover a minor sharp edge. In (f) FMM received higher scores due to the fact that it excluded surrounding tissue that shouldn't be part of the segmentation.



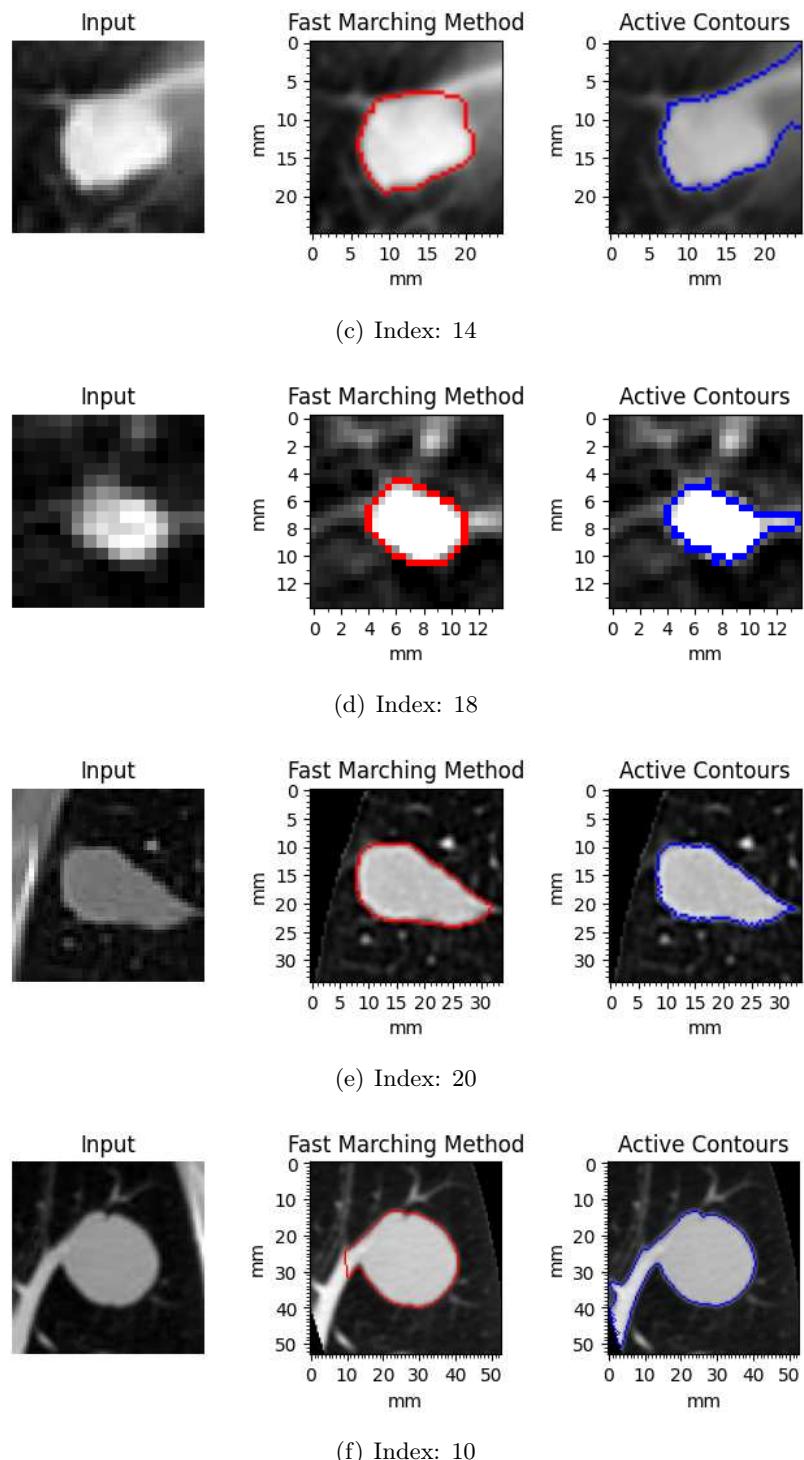


Figure 3.11: Cases where FMM and AC scored differently

### 3.2.3.2 Irregular

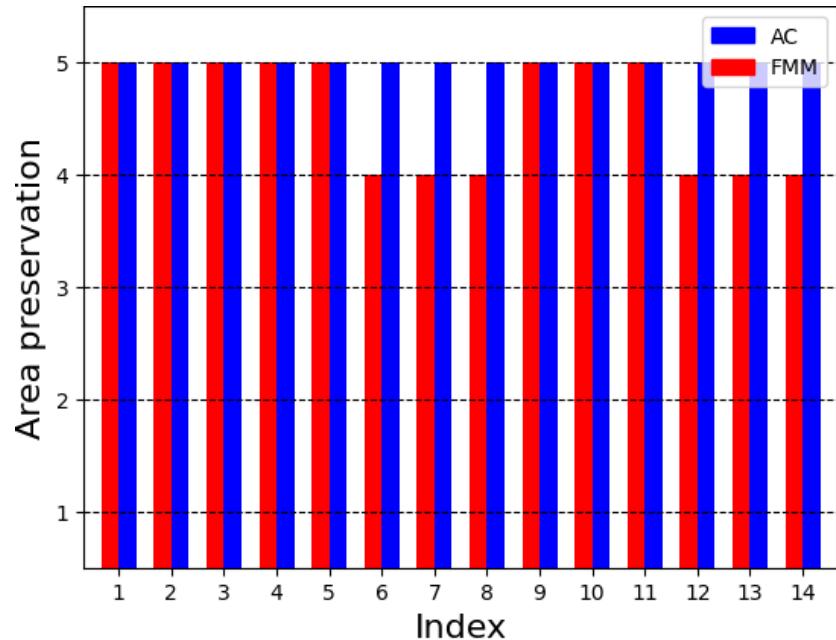


Figure 3.12: Irregular - Area preservation (1/2)

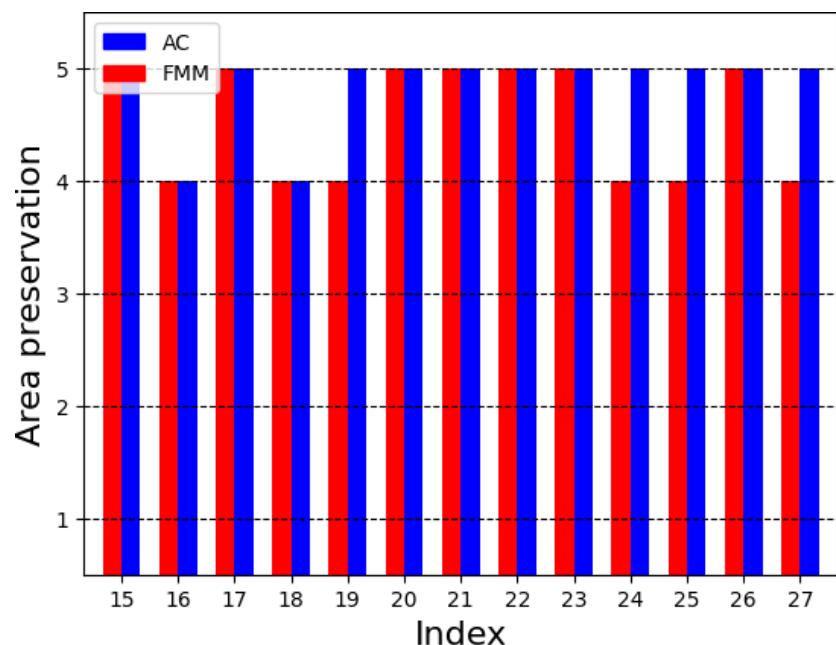


Figure 3.13: Irregular - Area preservation (2/2)

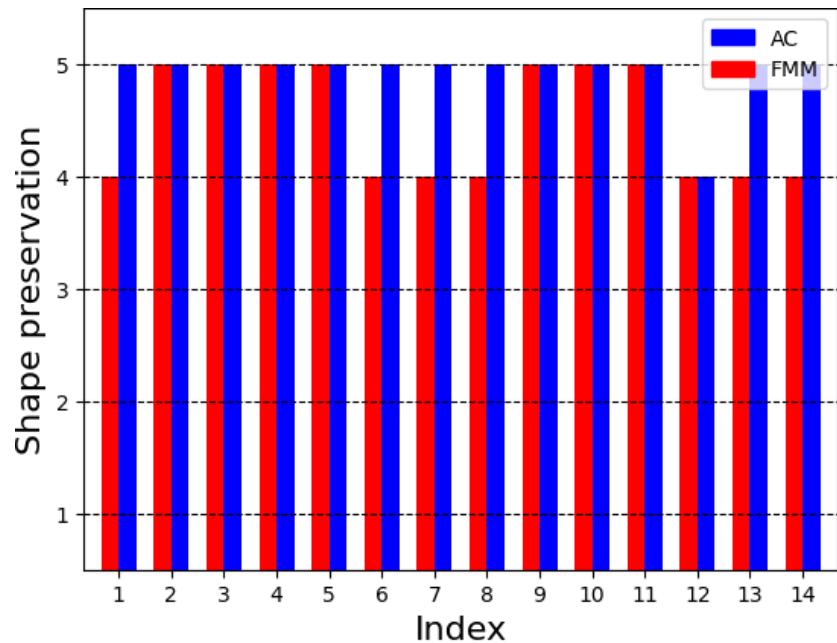


Figure 3.14: Irregular - Shape preservation (1/2)

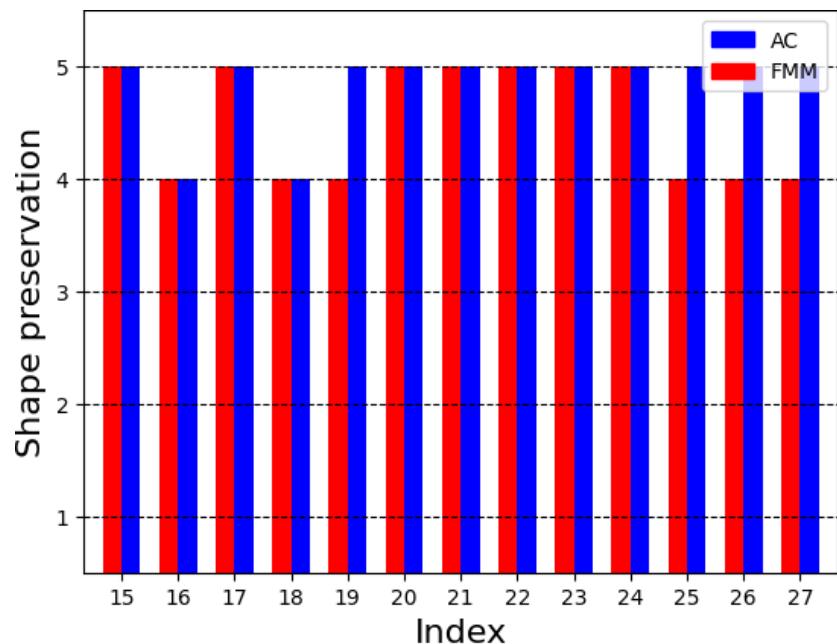


Figure 3.15: Irregular - Shape preservation (2/2)

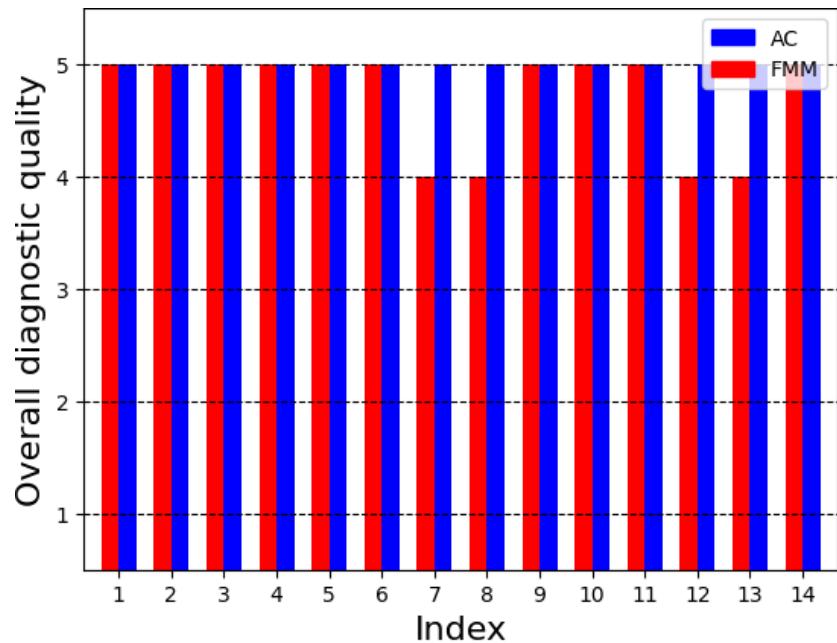


Figure 3.16: Irregular - Overall diagnostic quality (1/2)

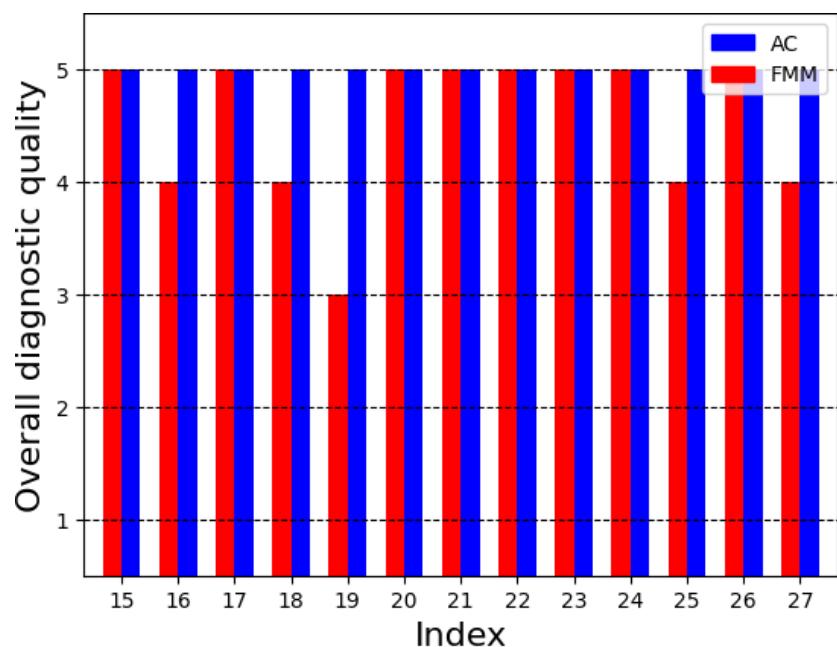


Figure 3.17: Irregular - Overall diagnostic quality (2/2)

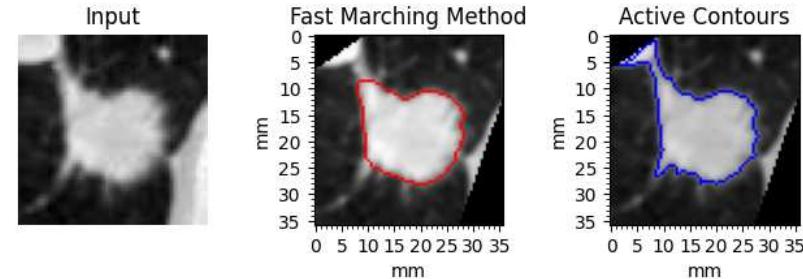
To summarize the results and make a comparison between the two methods, for each score the mean and standard deviation have been calculated and are shown in table 3.2.

	Area preservation	Shape preservation	Overall diag. qual.
FMM	$4.55 \pm 0.49$	$4.51 \pm 0.49$	$4.62 \pm 0.55$
AC	$4.92 \pm 0.26$	$4.88 \pm 0.31$	$5.0 \pm 0.0$

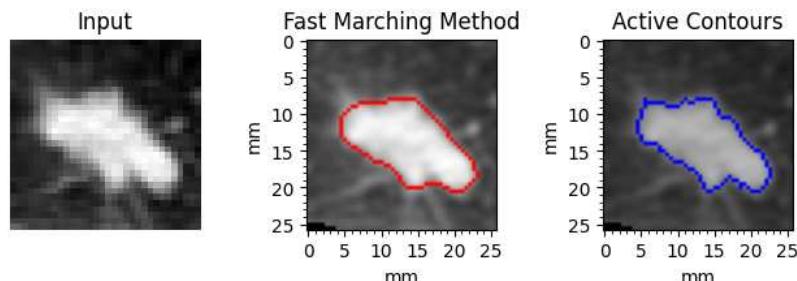
Table 3.2: Irregular subjective evaluation results summary

Although these types of nodules have an irregular border and are more difficult to segment than the previous type, both segmentation algorithms obtain excellent results with high mean average scores. Similarly to the previous case, AC has a slightly higher mean for all three scores.

There are several cases where AC scores higher than FMM by one point. In Figure 3.18 are the instances where AC rated higher than FMM on all three scores. In these cases AC gives a more detailed segmentation because it better captures the sharp edges present in irregular nodules, on the other hand FMM tends to give smoother more convex shaped results.



(a) Index: 7



(b) Index: 8

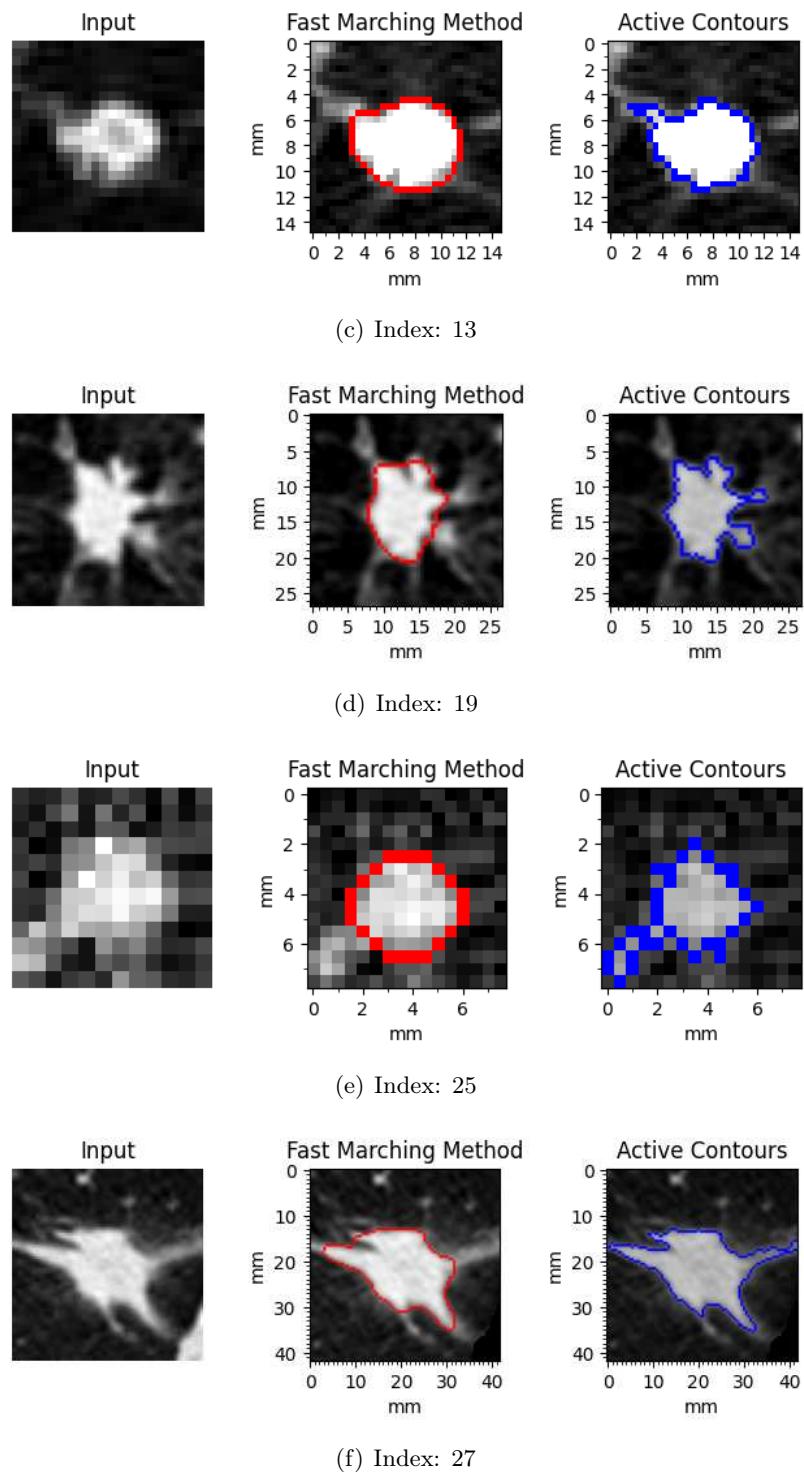


Figure 3.18: Cases where AC obtained a better rating

### 3.2.3.3 Semi-transparent

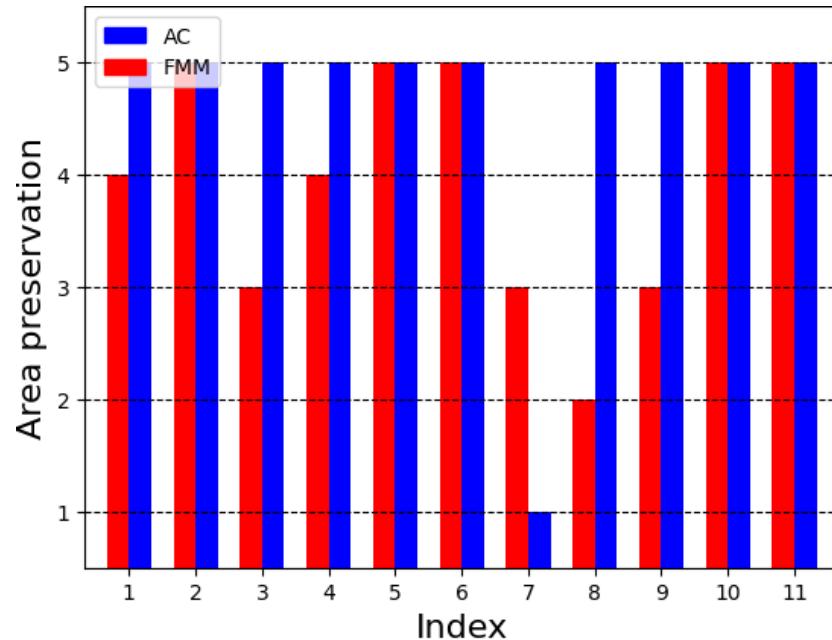


Figure 3.19: Semi-transparent - Area preservation (1/2)

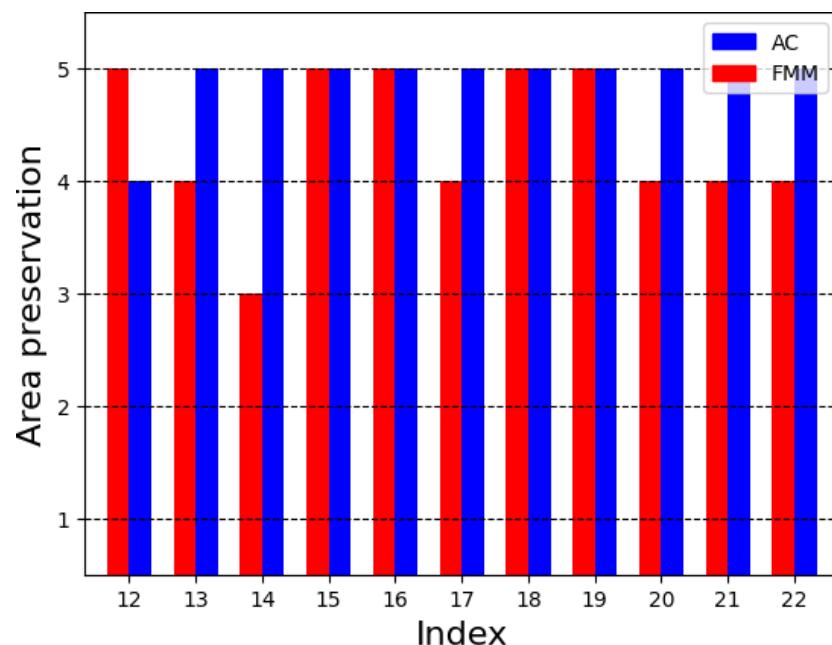


Figure 3.20: Semi-transparent - Area preservation (2/2)

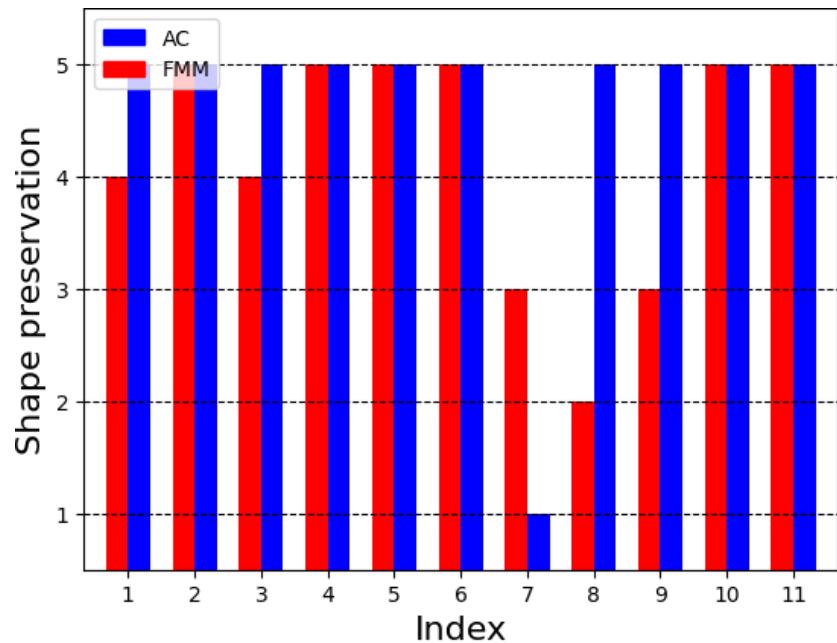


Figure 3.21: Semi-transparent - Shape preservation (1/2)

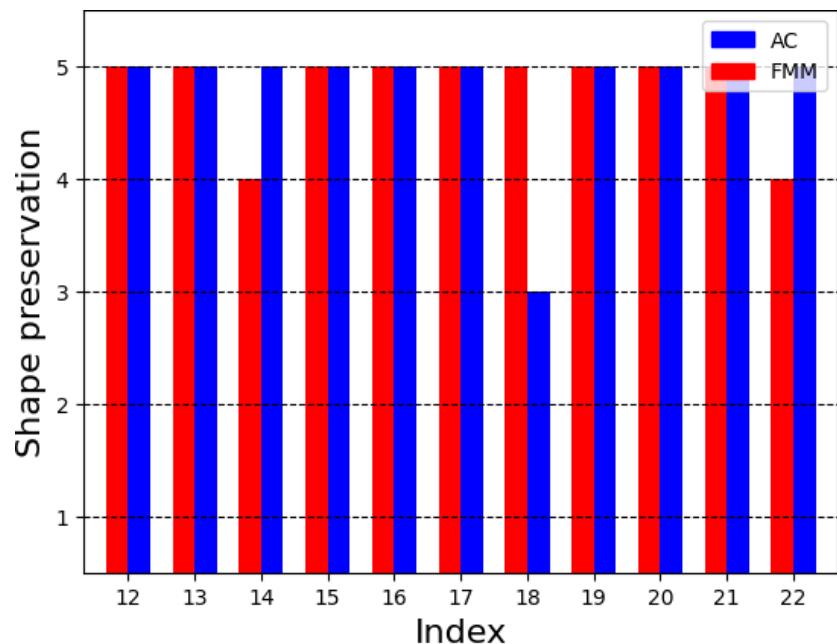


Figure 3.22: Semi-transparent - Shape preservation (2/2)

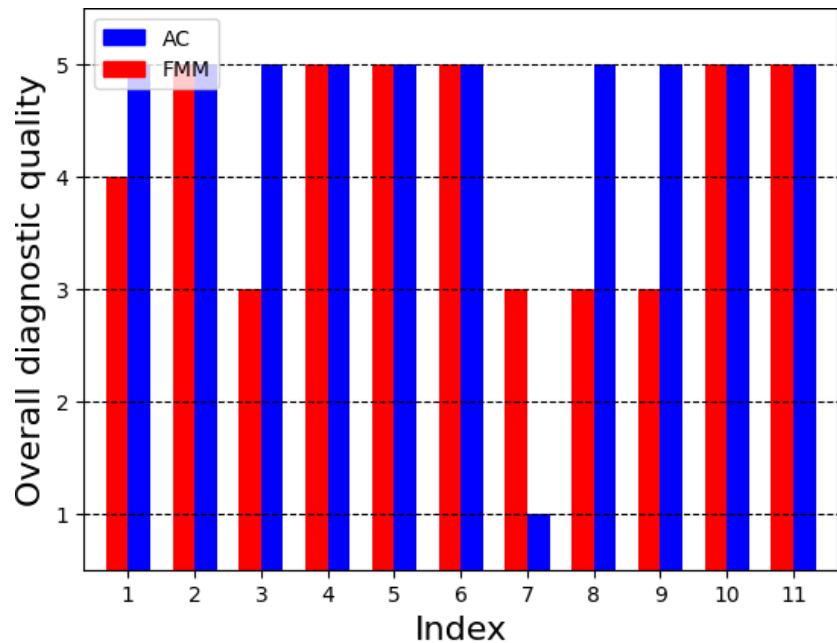


Figure 3.23: Semi-transparent - Overall diagnostic quality (1/2)

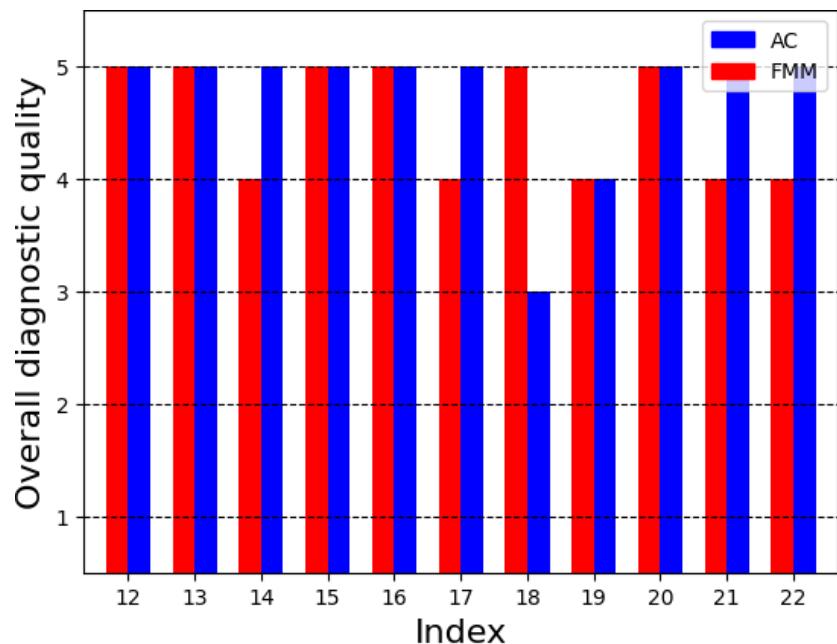


Figure 3.24: Semi-transparent - Overall diagnostic quality (2/2)

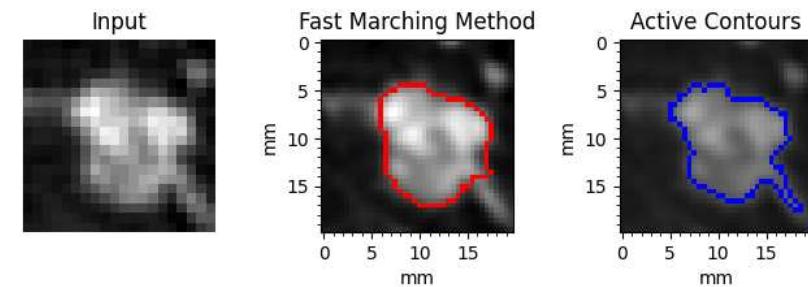
To summarize the results and make a comparison between the two methods, for each score the mean and standard deviation have been calculated and are shown in table 3.3.

	Area preservation	Shape preservation	Overall diag. qual.
FMM	$4.18 \pm 0.88$	$4.5 \pm 0.83$	$4.36 \pm 0.77$
AC	$4.77 \pm 0.84$	$4.72 \pm 0.91$	$4.68 \pm 0.92$

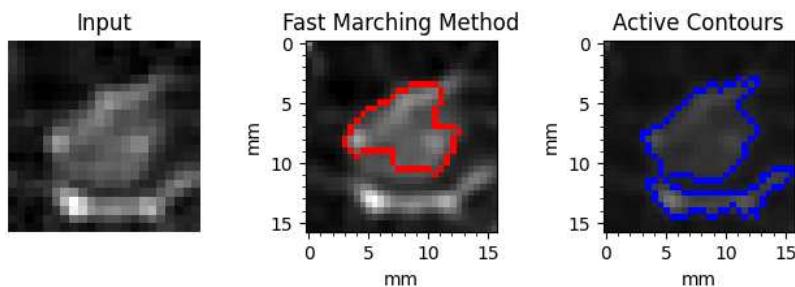
Table 3.3: Semi-transparent subjective evaluation results summary

Due to the non-homogeneous intensity and part-solid components these nodules can prove challenging when it comes to segmentation, nonetheless both segmentation algorithm received high scores, especially AC.

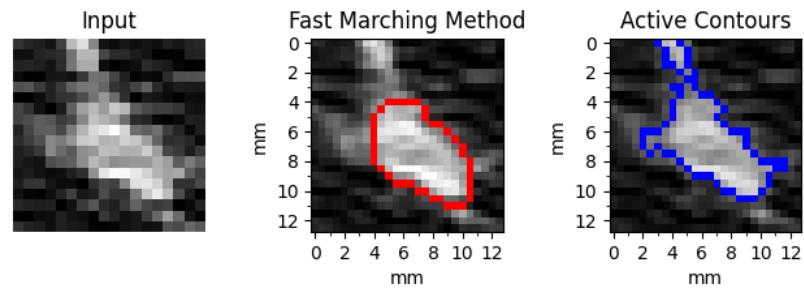
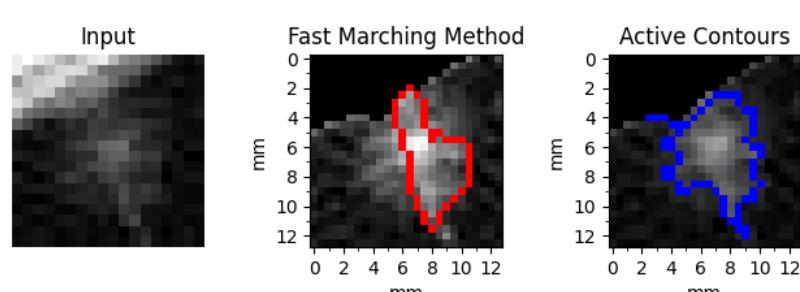
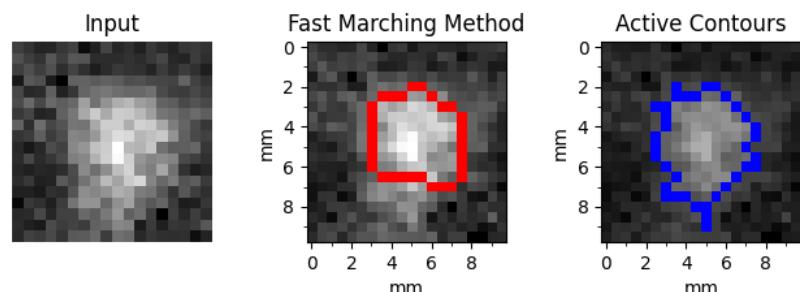
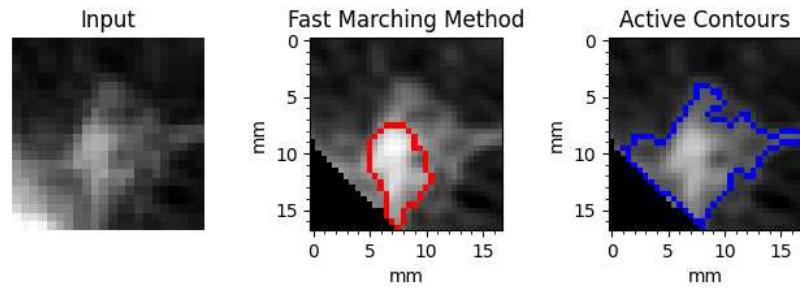
The cases where there is a difference in score are shown in Figure 3.25. In (a-g) AC was rated higher on all three scores, mostly due to the algorithm including more of the non-solid part of the nodule in the segmentation. In (h-j) are cases where FMM obtains a higher rating, in (h) AC fails completely because it mistakes a blood vessel for the nodule.



(a) Index: 1



(b) Index: 3



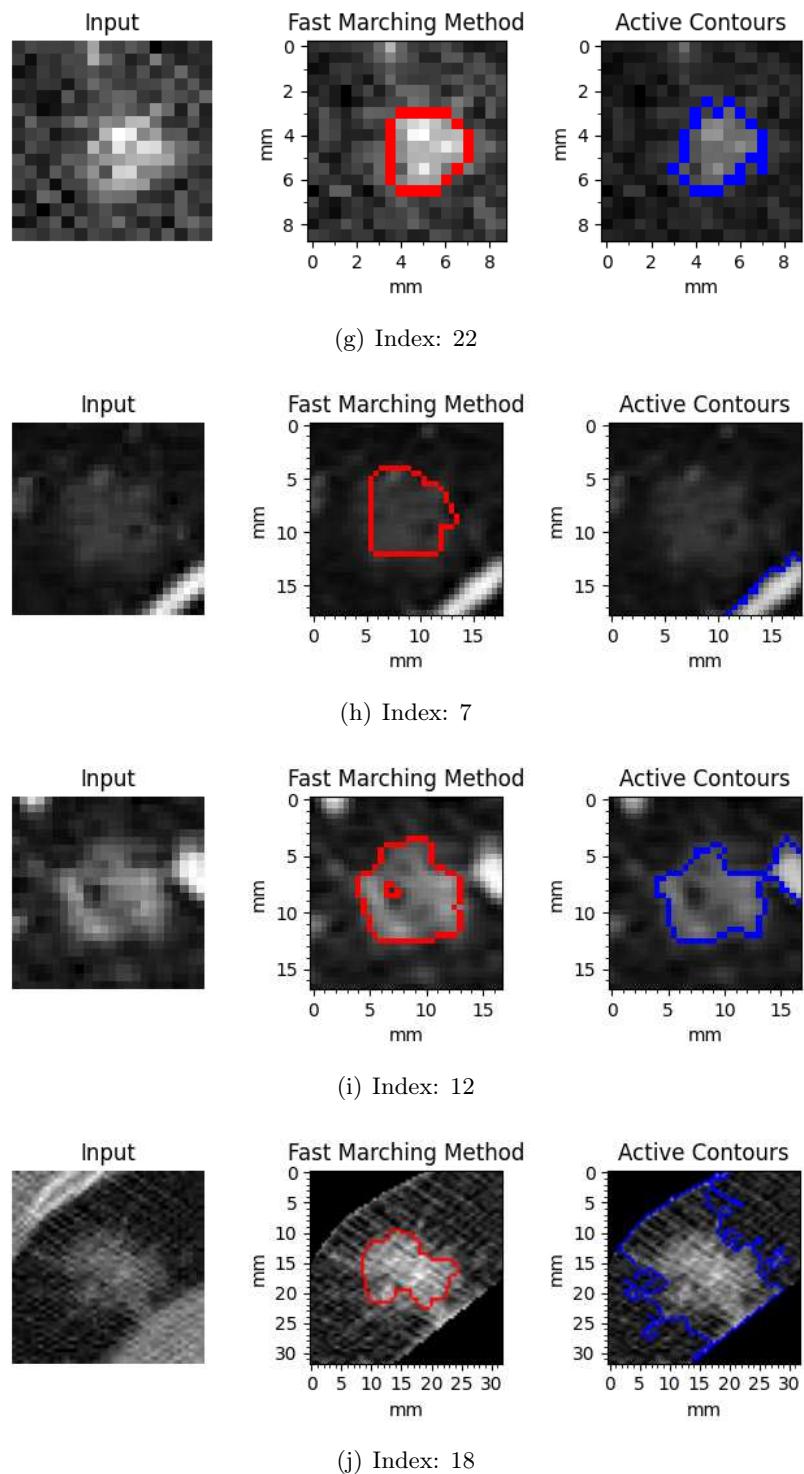


Figure 3.25: Cases where FMM and AC scored differently

### 3.2.3.4 Cavitary

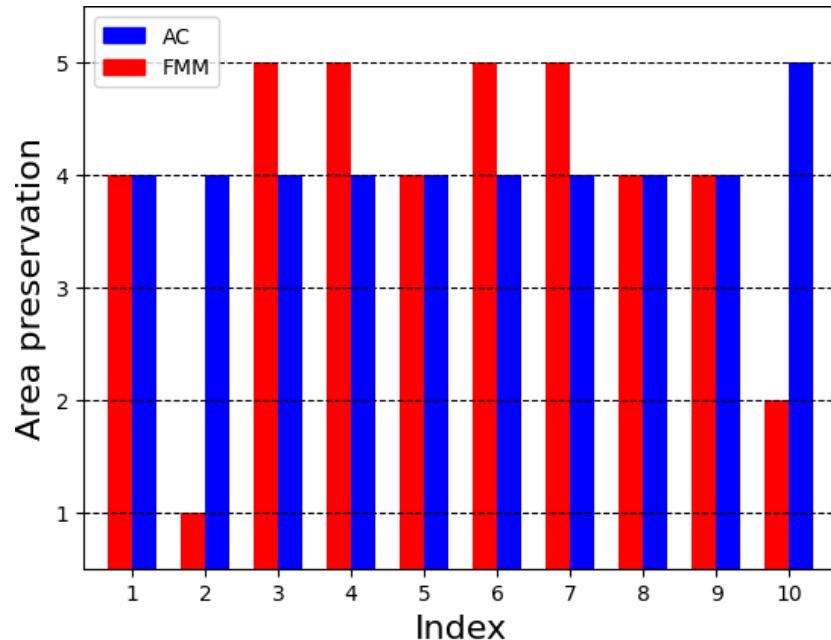


Figure 3.26: Cavitary - Area preservation

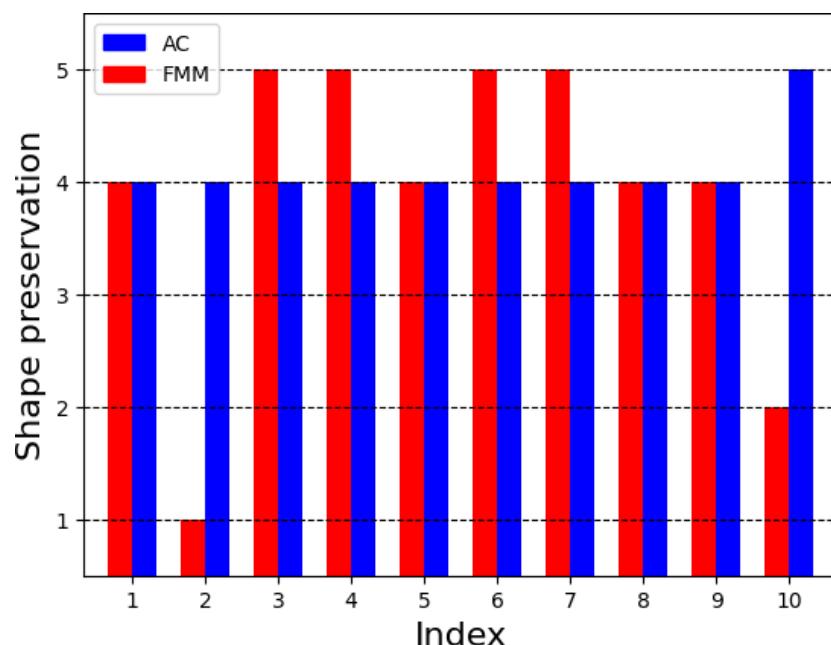


Figure 3.27: Cavitary - Shape preservation

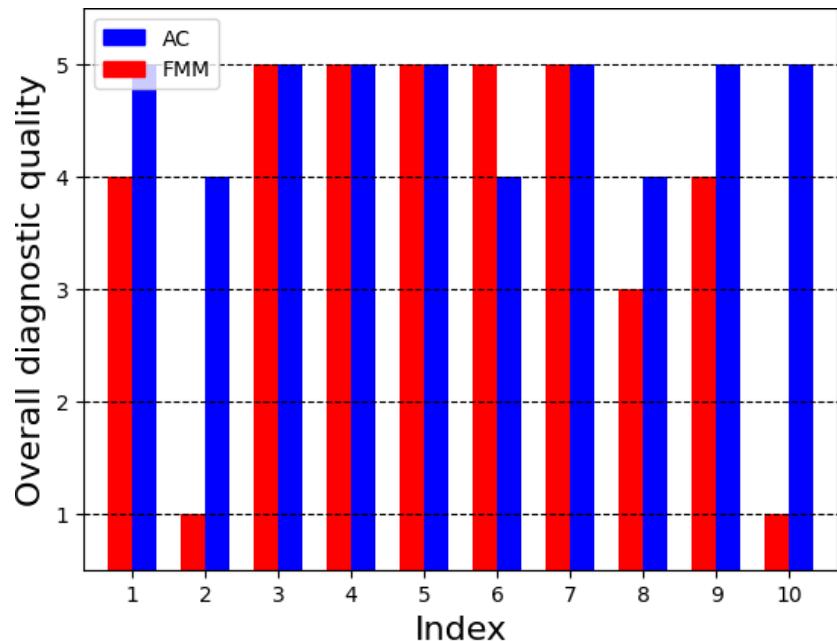


Figure 3.28: Cavitary - Overall diagnostic quality

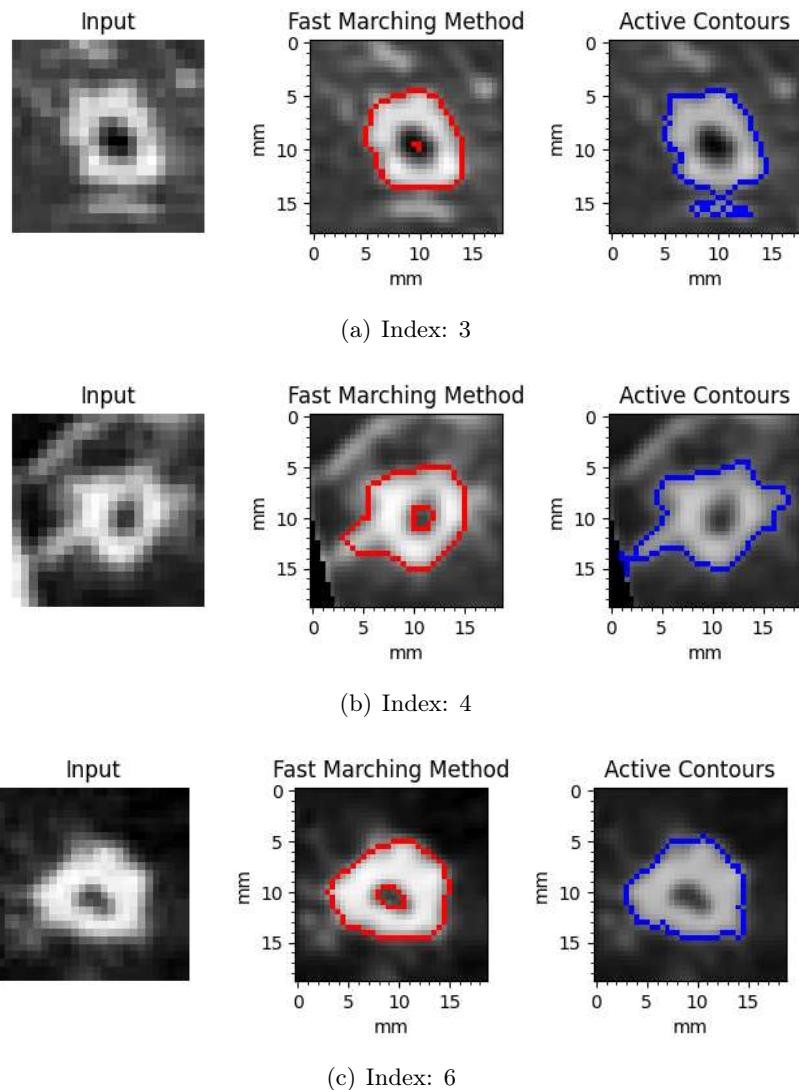
To summarize the results and make a comparison between the two methods, for each score the mean and standard deviation have been calculated and are shown in table 3.4.

	Area preservation	Shape preservation	Overall diag. qual.
FMM	$3.9 \pm 1.3$	$3.9 \pm 1.3$	$3.8 \pm 1.53$
AC	$4.1 \pm 0.3$	$4.1 \pm 0.3$	$4.7 \pm 0.45$

Table 3.4: Cavitary subjective evaluation results summary

The presence of a cavity makes these nodules particularly difficult to segment. Both algorithms obtained good results, but in terms of average scores AC was rated higher. This is due FMM failing completely in two instances, thus significantly lowering the mean.

The cases where there is a difference in score are shown in Figure 3.29. In (a-d) FMM was rated higher, in (a-c) FMM takes the cavity into account and the ring is sufficiently thick for the algorithm to work well. In (e-f) are the two instances where AC earner higher scores and FMM failed, this is largely due to the ring being too thin for the algorithm to properly cluster and merge.



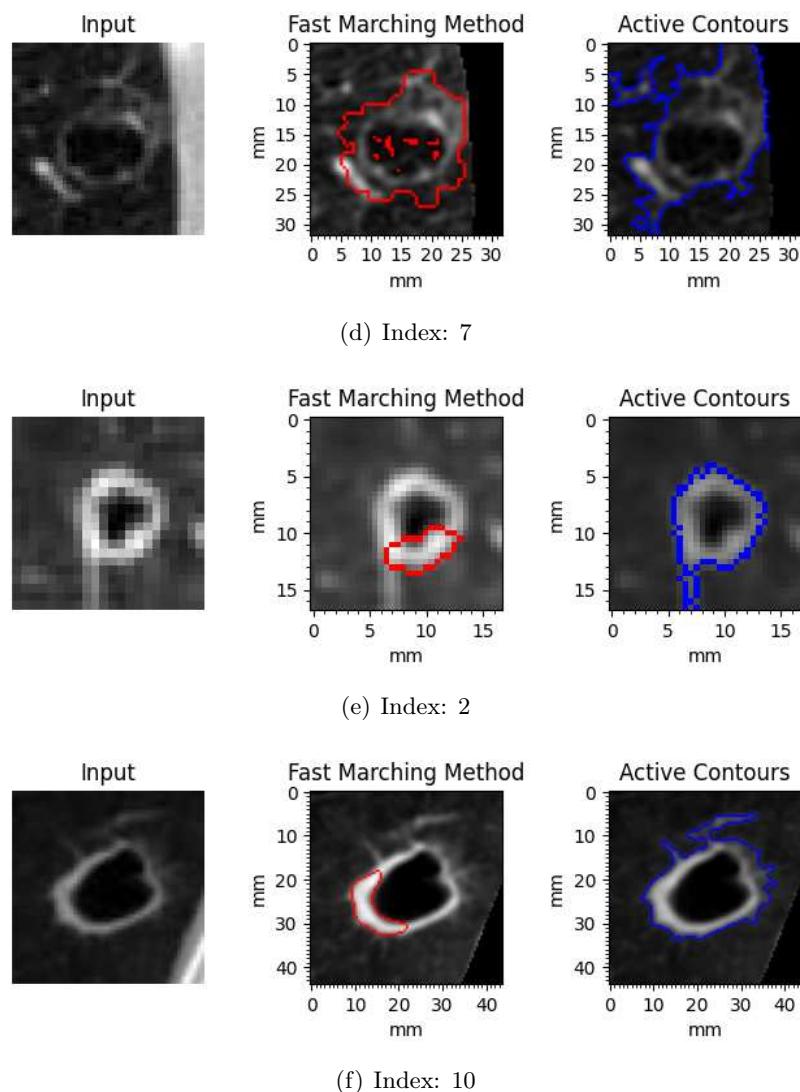


Figure 3.29: Cases where FMM and AC scored differently

### 3.3 Objective evaluation

As discussed earlier, two datasets and methods of evaluation are used to check whether the results of both methods are consistent, and to inspect the inconsistencies if there are any.

Since perfectly circular isolated nodules are too easy to segment, the phantom data was altered to obtain nodules with more complex shapes and other additions to make them resemble real nodules. For evaluation two different metrics were used: Intersection over Union (IoU) and diameter relative error. This will be elaborated in the following sections.

#### 3.3.1 Dataset construction

This dataset is entirely constructed by modifying the CT data from the lung phantom scans. The advantage of using a phantom is having a more controlled and homogeneous environment to test the segmentation on, and compared to real patient data the artificial nodules have a predetermined size and shape. This way the ground truth images can be easily obtained, and there is no need for radiologist annotations. Spherical artificial nodules of three different sizes (5mm, 10mm and 20mm) have been placed in a CT scanner and four different scans were performed, with various tube currents. Higher tube currents give improved image quality and decreased image noise, but raise the radiation dose, which is not an issue when dealing with a phantom. The scans were performed with tube currents of 30mA, 60mA, 99mA and 197mA. The rest of the parameters were the same for every scan and can be seen in Table 1.1.

The issue with using this data to evaluate the segmentation is the lack of complexity, which was added by modifying the nodule images. These alterations to the data have been performed by using tools from the GNU Image Manipulation Program (GIMP) [62]. The process is described in the following steps:

- (a) The desired boundary is selected using the "Free Selection" tool, this way the nodule shape can be selected at will and the nodule image and ground truth will be made to fit this contour.
- (b) To fit the nodule to the newly selected boundary the "Warp Transform" tool is used, which applies an image warping transformation with a brush. For the foreground the warping is applied from the inside to the outside, and for the background it is applied from outside to the inside.
- (c) The ground truth image is modified to fit the selected boundary by simply painting the inside of the selection white and the outside black.

- (d) Warping leaves the undesired effect of having strong edges along the boundary, this is corrected by using the "Smudge" tool, which uses a brush to smudge colors, it takes color in passing and uses it to mix it to the next colors it meets. This results in softer and more realistic edges along the boundary.

An example of each step is shown in Figure 3.30, where a 20mm nodule from the phantom is turned into an altered nodule belonging to the round category. For irregular type nodules during the selection step a more irregular shape is created, and the rest of the procedure is the same.

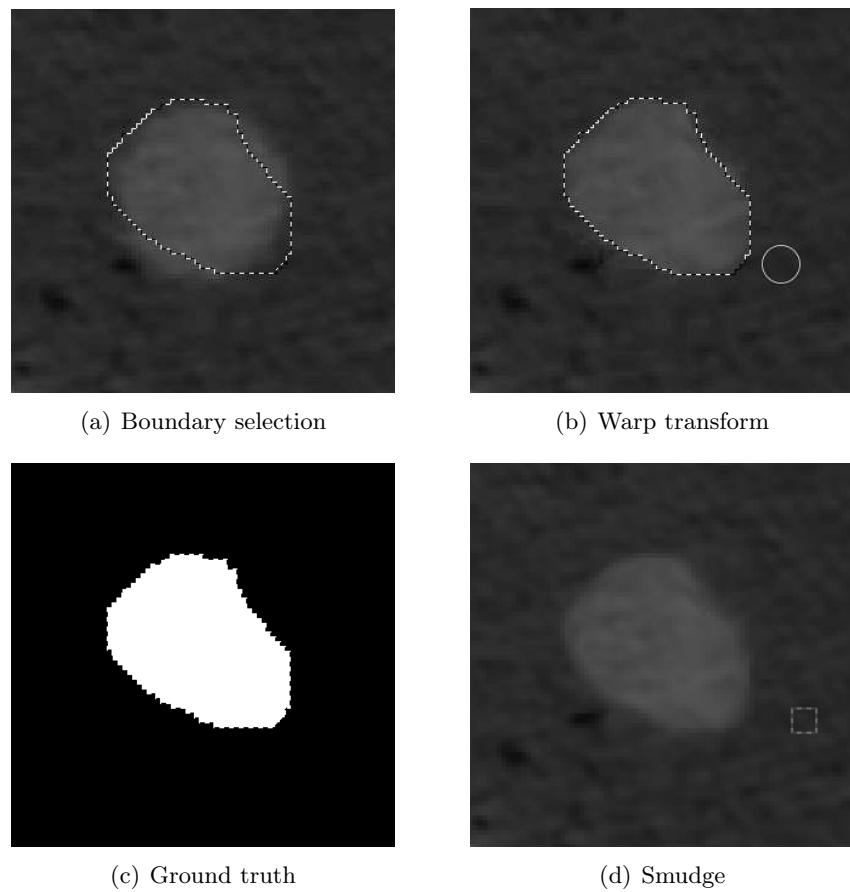


Figure 3.30: Creating a round artifical nodule

The previous method was used for creating nodules of the round and irregular type. For cavitary nodules a few more steps are needed:

- (a) With the "Free Selection" tool the shape of the cavity is selected.
- (b) Using the "Burn" tool, which darkens the pixels selected with a brush, the inside of the cavity is darkened.
- (c) The ground truth image is modified to exclude the cavity by painting it black.
- (d) The "Smudge" tool is applied to make the edges along the cavity softer.

An example of the added steps needed to create an artificial cavitary nodule is shown in Figure 3.31

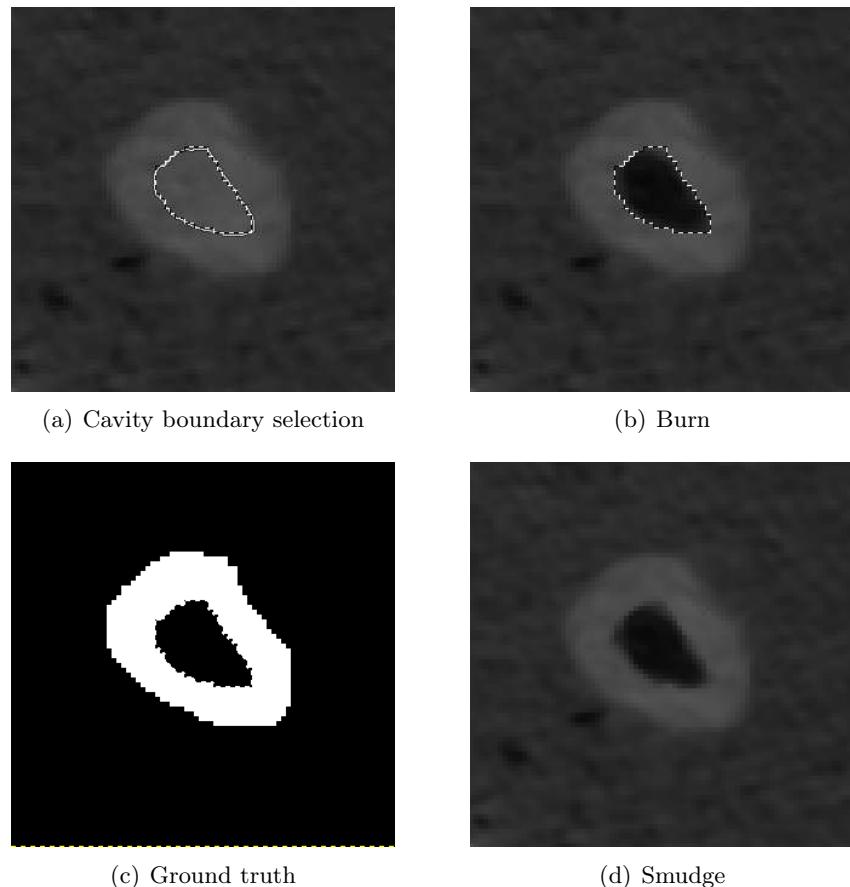


Figure 3.31: Creating a cavitary artificial nodule

For nodules of the semi-transparent type another method is used. The base procedure is the same as for the previous types, with the addition of a couple of steps. An example is shown in Figure 3.32. The following steps are applied:

- (a) Using the "Dodge" ( lightens pixels ) and "Burn" ( darkens pixels ) tools, the pixel intensities inside the nodule boundary are altered to make it appear similar to a ground-glass opacification (GGO).
- (b) With the "Smudge" tool the pixel colors inside the nodule are mixed to make it appear more lifelike .

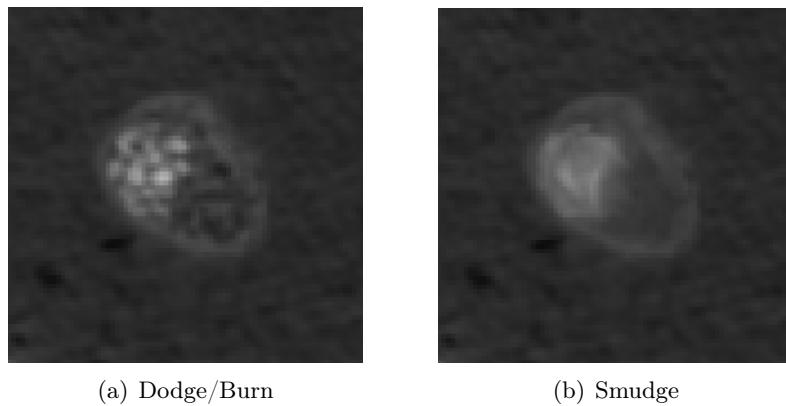


Figure 3.32: Creating a semi-transparent artificial nodule

To add more variety and make the dataset more representative, juxtaplureal and juxtavascular nodules are added. The process for creating juxtrapleural nodules follows these steps:

- (a) The nodule image is cropped to a convex shape similar to a lung wall, using the "Free Selection" tool and painting the outside of the selection black.
- (b) The ground truth image is updated to exclude the outside of the simulated lung wall.

An example of the procedure is shown in Figure 3.33.

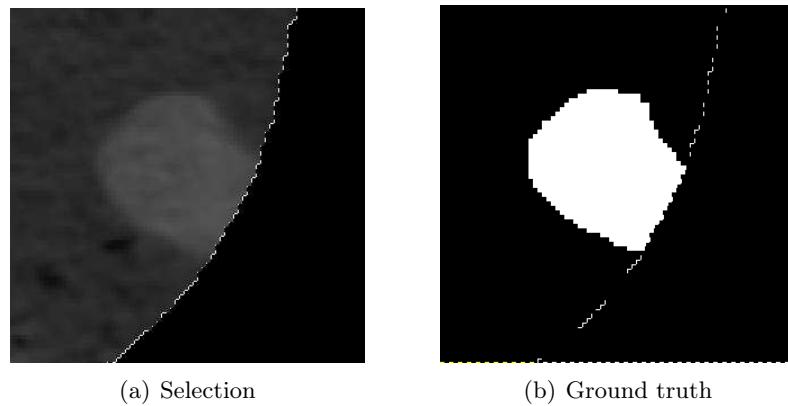


Figure 3.33: Creating a juxtapleural artificial nodule

To add blood vessels to the nodules following process is applied:

- Simulated blood vessels are added to the nodule by lightening that area of the image using the "Dodge" tool.
- Afterwards the edges are smoothed using the "Smudge" tool.

An example of the procedure is shown in Figure 3.34

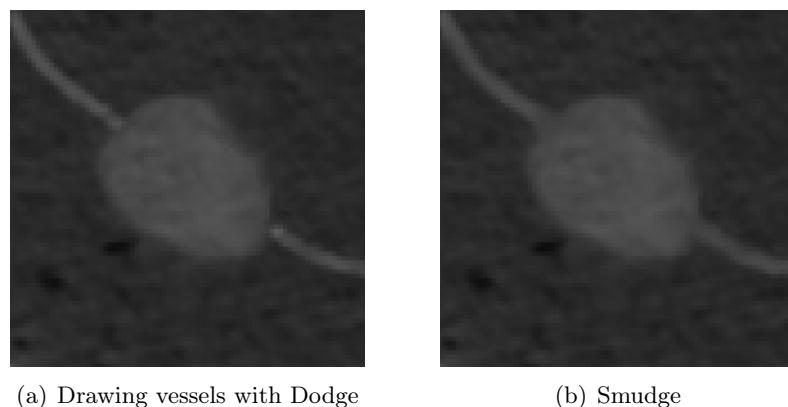


Figure 3.34: Creating a juxtavascular artificial nodule

At last by using these methods a dataset is created containing 108 nodules of different sizes, shapes, surroundings and tube currents. The nodules are organised in four different categories: round, irregular, semi-transparent and cavitary. The nodules are categorised in an analogous way as the ones from the LIDC subset. In each category containing altered nodules there are also juxtapleural and juxtravascular cases. Having readied the dataset, the nodules can be segmented and evaluated.

### 3.3.2 Evaluation method

Both segmentation algorithms are applied on the nodules from the phantom dataset, and the resulting binary masks are compared to the ground truth. The performance evaluation is done via two metrics:

- Intersection over Union : this metric quantifies the overlapping area between the segmented mask and ground truth
- Relative error of long-axis and short-axis diameter measurements : this metric calculates the difference between the measured and expected lengths of the nodule.

#### Intersection over Union

The first evaluation metric used is Intersection over Union, it can be interpreted as a similarity measure between a finite number of sets. For two sets A and B, it is defined as shown in Equation 3.3.1.

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3.3.1)$$

Having two binary images, A and B become two matrices containing ones and zeros. The intersection can be obtained as an element wise multiplication between the two matrices ( using `numpy.multiply` ). The union is obtained as the sum of A and B minus the intersection. To calculate the areas from the binary matrices it is sufficient to calculate the sum of all elements in the matrices ( using `numpy.sum` ).

An example is shown in Figure 3.35, in this case the intersection area is equal to 28 pixels and the union area is equal to 87 pixels, so the IoU score becomes  $28/87 = 0.322$ .

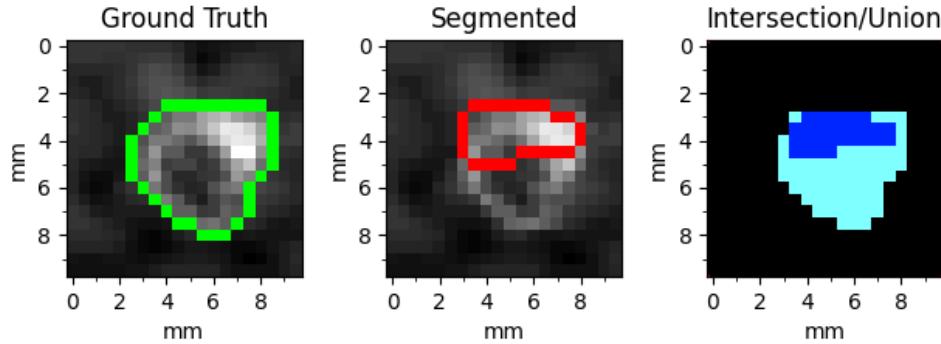


Figure 3.35: Intersection over Union example

### Long-axis and short-axis diameter relative error

With this evaluation metric, the measured lengths of the ground truth is compared to the measured lengths of the segmentation. Manual diameter measurements are the most widely used technique for quantifying the size of lung nodules in CT scans, this automated method will follow the same procedure. Following the "Recommendations for Measuring Pulmonary Nodules at CT" from the Fleischner Society [63], the long-axis diameter is taken as the maximal distance from two points on the nodule border and the short-axis diameter is taken as the maximum distance that is perpendicular to the long-axis. The values should be expressed to the nearest whole millimeter. These are usually done manually by radiologists, for this automated procedure an algorithm was written that performs the same measurements. The input is the binary image of a nodule and the outputs are the long-axis and short-axis diameter measures, the algorithm works as follows:

- Border extraction : since only the pixels on the border are of interest, they are added to a list and the rest are disregarded. This is done by applying binary erosion (using `ndimage.morphology.binary_erosion`) on the binary image and subtracting the eroded result from it. The coordinates of the remaining pixels are written to a list (using `numpy.where`).

- Long-axis measure : firstly the centroid of the binary image is calculated by using `skimage.measure.label` and `skimage.measure.regionprops`, these functions measure properties of the image region corresponding to the nodule, one of these properties is the centroid. To find the two most distant pixels on the border, at first the pixel that is the furthest from the centroid is found by comparing all the border pixel distances from the centroid and taking the maximum, this will be the first endpoint of the long-axis. To find the second endpoint, the distances of all border pixels are calculated from the first endpoint and the maximum is taken. By calculating the distance between the two endpoints the long-axis diameter is obtained.
- Short-axis measure : at first a set of perpendicular lines to the long-axis is constructed. The slope of the long-axis is calculated and also its negative reciprocal, which is the slope of a perpendicular line. Iterating over all possible lines joining two border pixels, a list is constructed by adding lines that are perpendicular to the long-axis. Since the coordinate system is discrete, instead of being perfectly perpendicular (the slope of the perpendicular line is equal to the inverse reciprocal of the long-axis slope) an approximation is taken.

The angle  $\theta$  between two lines with slopes  $m_1$  and  $m_2$  can be calculated using equation 3.3.2.

$$\tan(\theta) = \pm \frac{m_2 - m_1}{1 + m_1 \cdot m_2} \quad (3.3.2)$$

Two lines will be considered "almost" perpendicular if the  $\tan(\theta)$  calculated using the previous equation is smaller than 0.15, this value was chosen empirically and corresponds to an angle of  $\theta \approx 8.5^\circ$ . Edge cases where the slope is 0 or  $\infty$  are handled accordingly. Having a set of lines that are "almost" perpendicular to the long-axis, the longest one is taken as the short-axis measurement.

An example of this procedure is shown in Figure 3.36. At first the border pixels are singled out (a), afterwards the long-axis is calculated starting from the centroid (b), the set of "almost" perpendicular lines to the long-axis is constructed (c), and finally the short-axis is calculated (d).

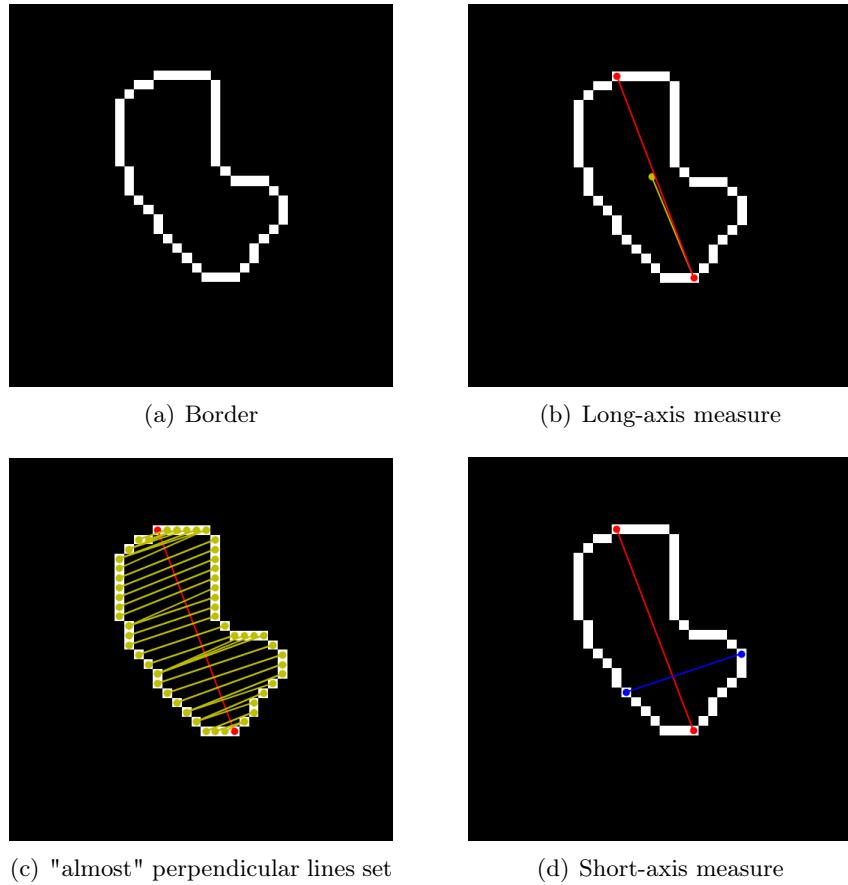


Figure 3.36: Measuring long-axis and short-axis diameter

Having devised a way to calculate long-axis and short-axis diameters, what remains is to compare the measurements of the segmentation result with the ground truth. To this end the relative error percentage is calculated for both axes, using the classical definition from equation 3.3.3, where  $m_s$  is the measurement obtained from the segmentation and  $m_{gt}$  is the measurement obtained from the ground truth.

$$\delta = \left| \frac{m_s - m_{gt}}{m_{gt}} \right| \cdot 100\% \quad (3.3.3)$$

### **3.3.3 Results**

In this section the results from the objective evaluation are displayed and discussed by category. For each metric a graph is shown and the numerical results are recapped in a table. Following is a discussion of the results with some key examples. All the segmentation instances with the outlined borders and related evaluation scores can be viewed in appendix. The indexing of nodules is kept consistent for all graphs, tables and figures.

### 3.3.3.1 Round

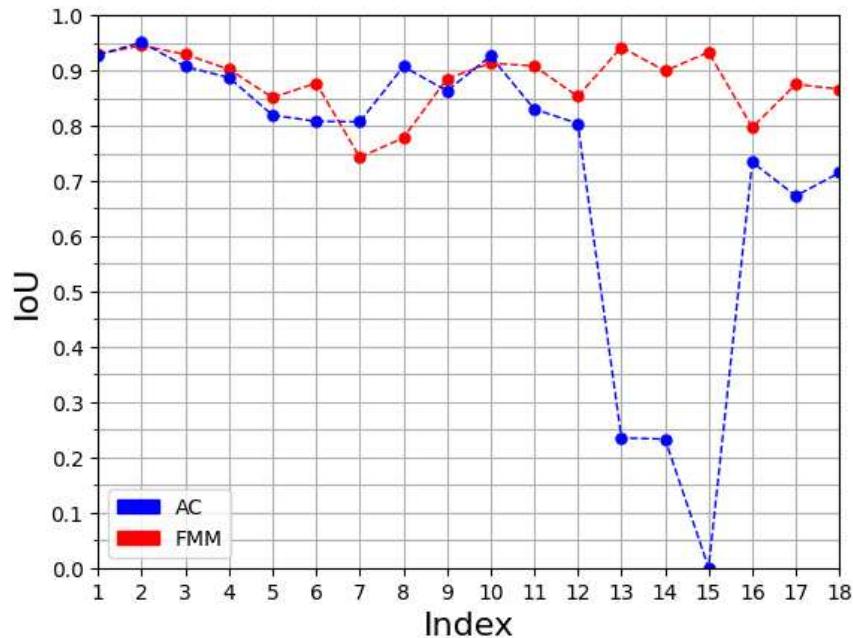


Figure 3.37: Round - Intersection over Union (1/2)

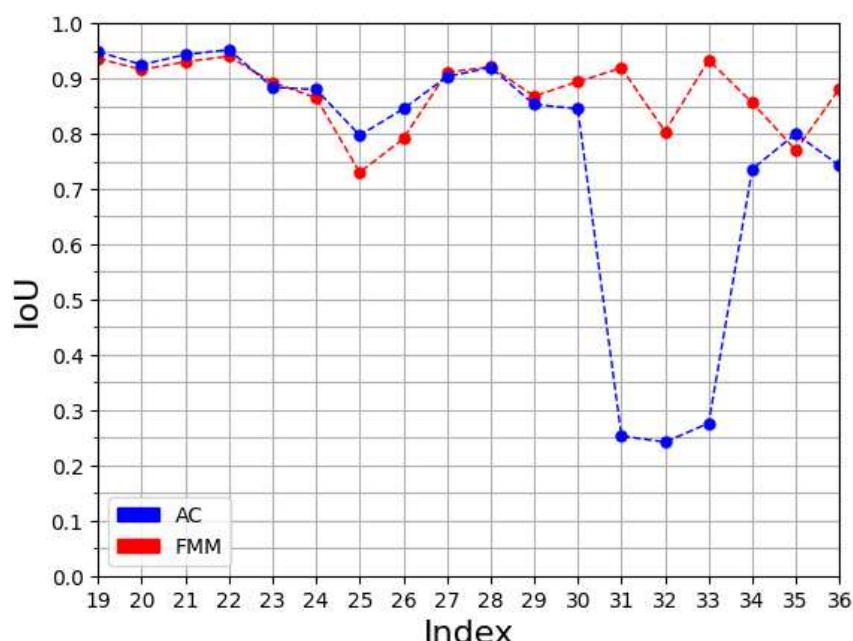


Figure 3.38: Round - Intersection over Union (2/2)

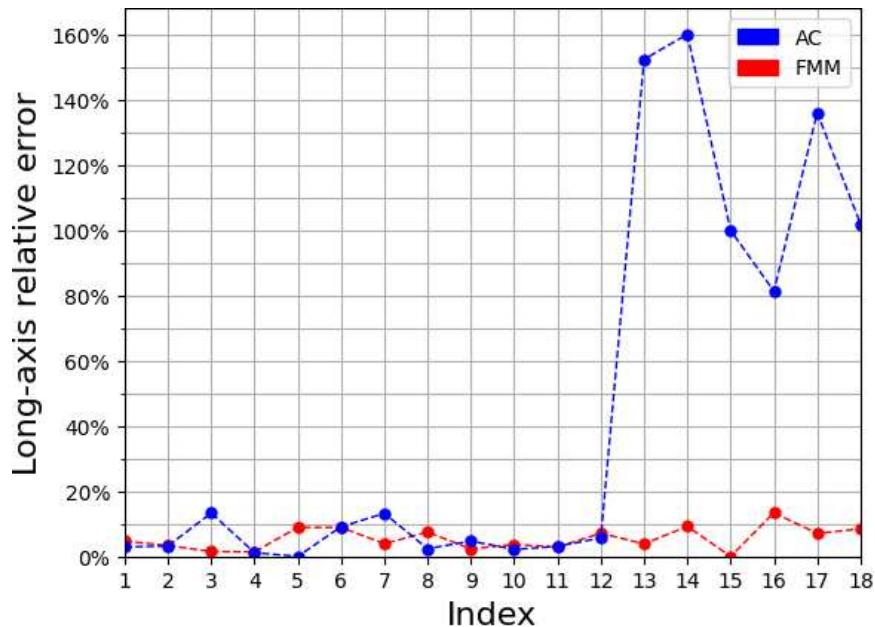


Figure 3.39: Round - Long-axis relative error (1/2)

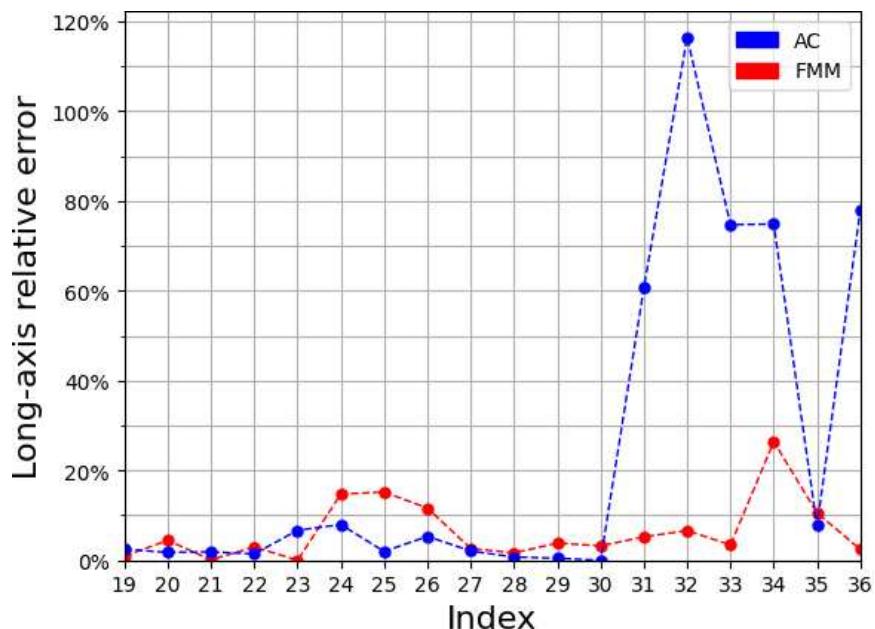


Figure 3.40: Round - Long-axis relative error (2/2)

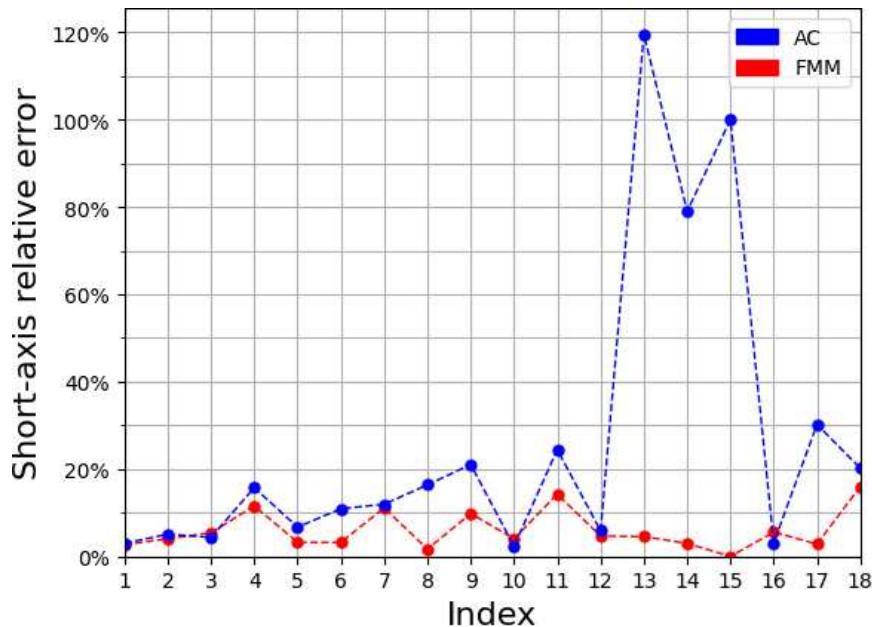


Figure 3.41: Round - Short-axis relative error (1/2)

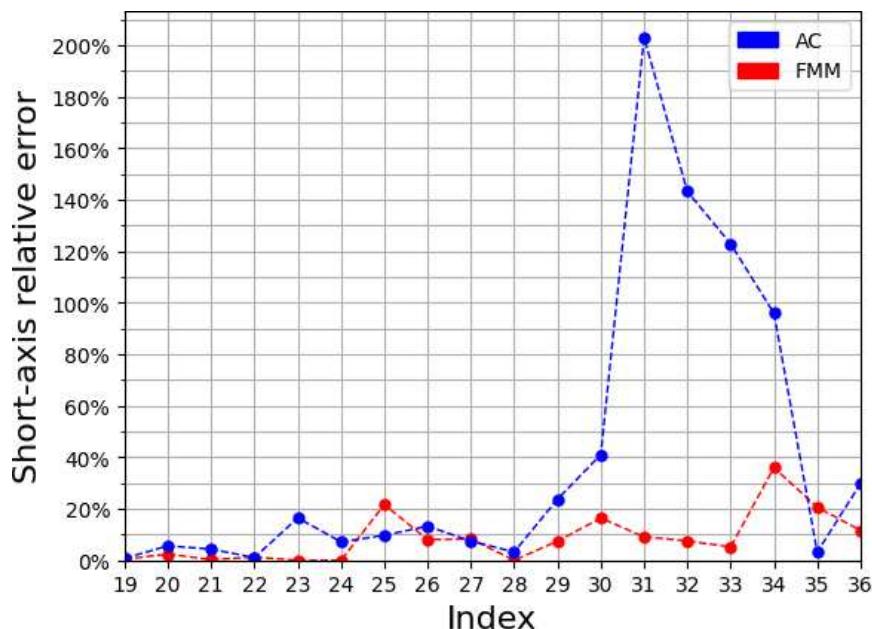


Figure 3.42: Round - Short-axis relative error (2/2)

To summarize the results and make a comparison between the two methods, for each metric the mean and standard deviation have been calculated and are shown in table 3.5. The standard deviation is high for both algorithms, especially for AC due to segmentation failure in juxtapleural cases.

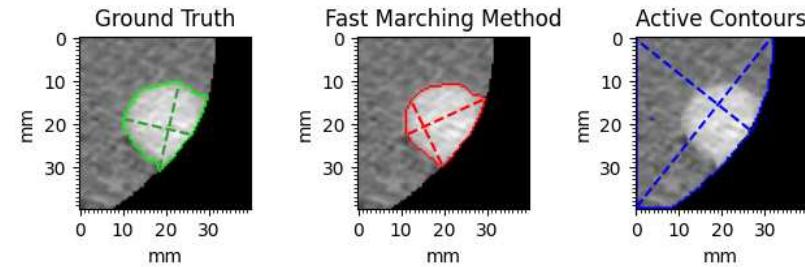
	IoU	Long-axis err.[%]	Short-axis err.[%]
FMM	$0.877 \pm 0.058$	$5.94 \pm 5.32$	$7.32 \pm 7.48$
AC	$0.743 \pm 0.252$	$34.4 \pm 49.2$	$33.6 \pm 48.0$

Table 3.5: Round objective evaluation results summary

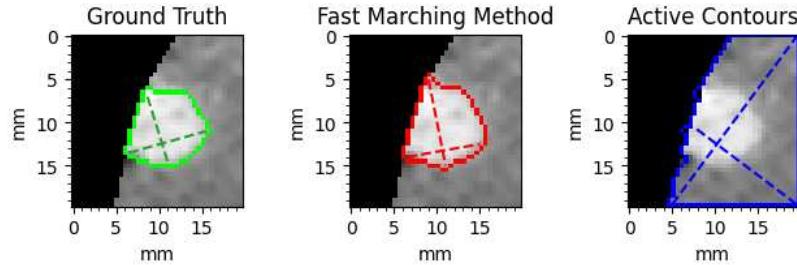
In this case the fast marching method obtained superior scores for all metrics and overall works better than active contours for segmenting these type of nodules. FFM can handle these round type nodules well because they are solid with a convex shape and sufficiently regular border. Additionally it performs better than AC in juxtapleural and juxtapleural cases.

Long-axis and short-axis diameter errors are in line with the IoU scores, due to the sufficiently regular border of this type of nodule, meaning that area overlap is consistent with diameter measures.

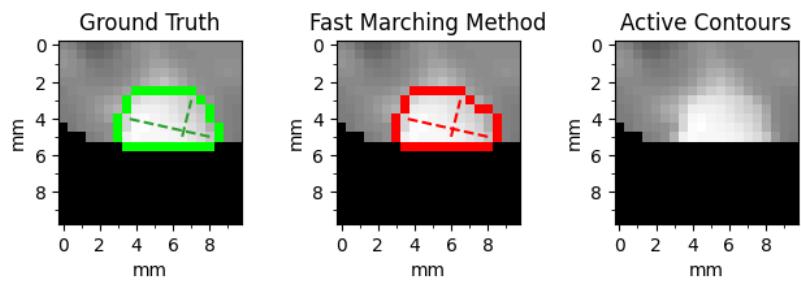
Considering an IoU less than 0.5 to correspond to a failed segmentation, the cases where this happens are shown in Figure 3.43, they all correspond to active contour segmentations which can have difficulties when segmenting juxtapleural nodules.



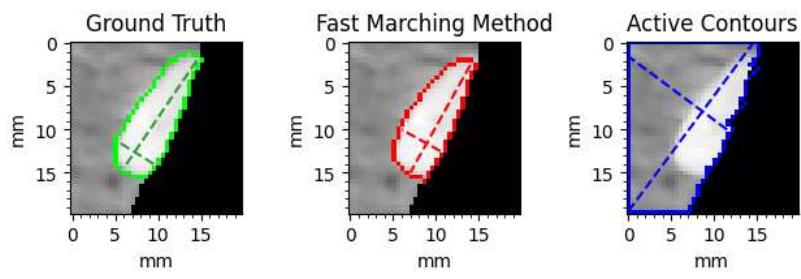
(a) Index: 13



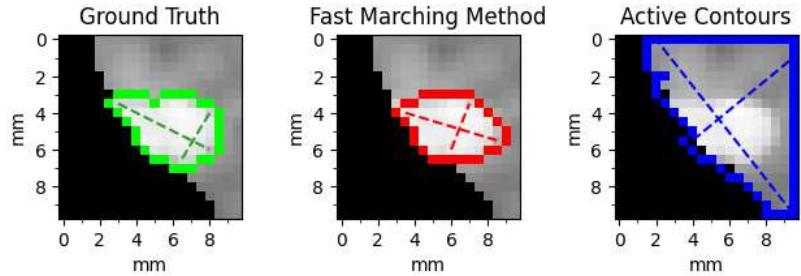
(b) Index: 14



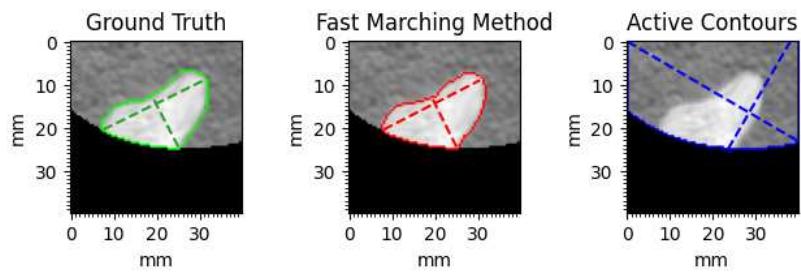
(c) Index: 15



(d) Index: 31



(e) Index: 32



(f) Index: 33

Figure 3.43: IoU < 0.5

In the rest of the cases the segmentation gives accurate results. In Figure 3.44 are some examples where FMM outperforms AC. When it comes to juxtapavascular nodules, FMM performs better because it excludes the vessels and only segments the nodule (b,d).

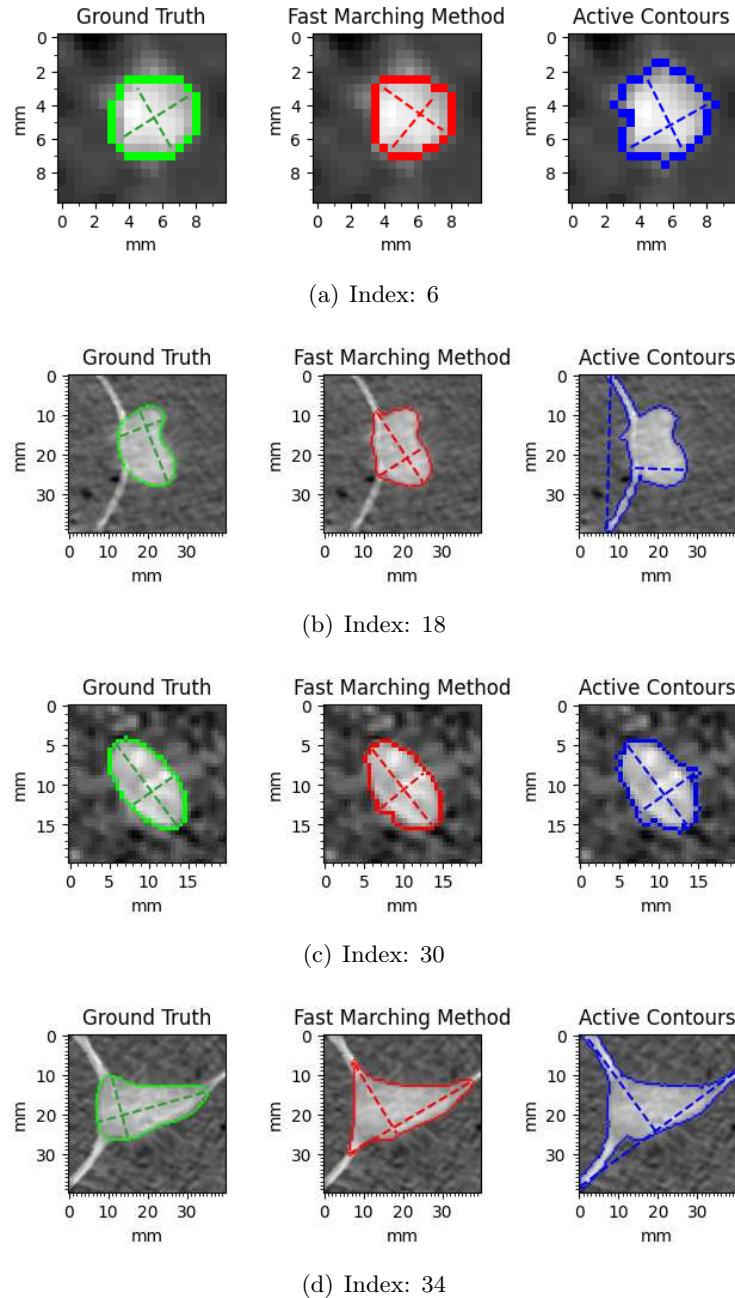


Figure 3.44: FMM better than AC ( $\text{IoU} > 0.5$ )

And in Figure 3.45 are some cases where AC segmentation achieves the better IoU score. Mostly with small nodule sizes, but this doesn't correlate to better performance with smaller size, because there are just as many cases where FMM performs better with small nodules.

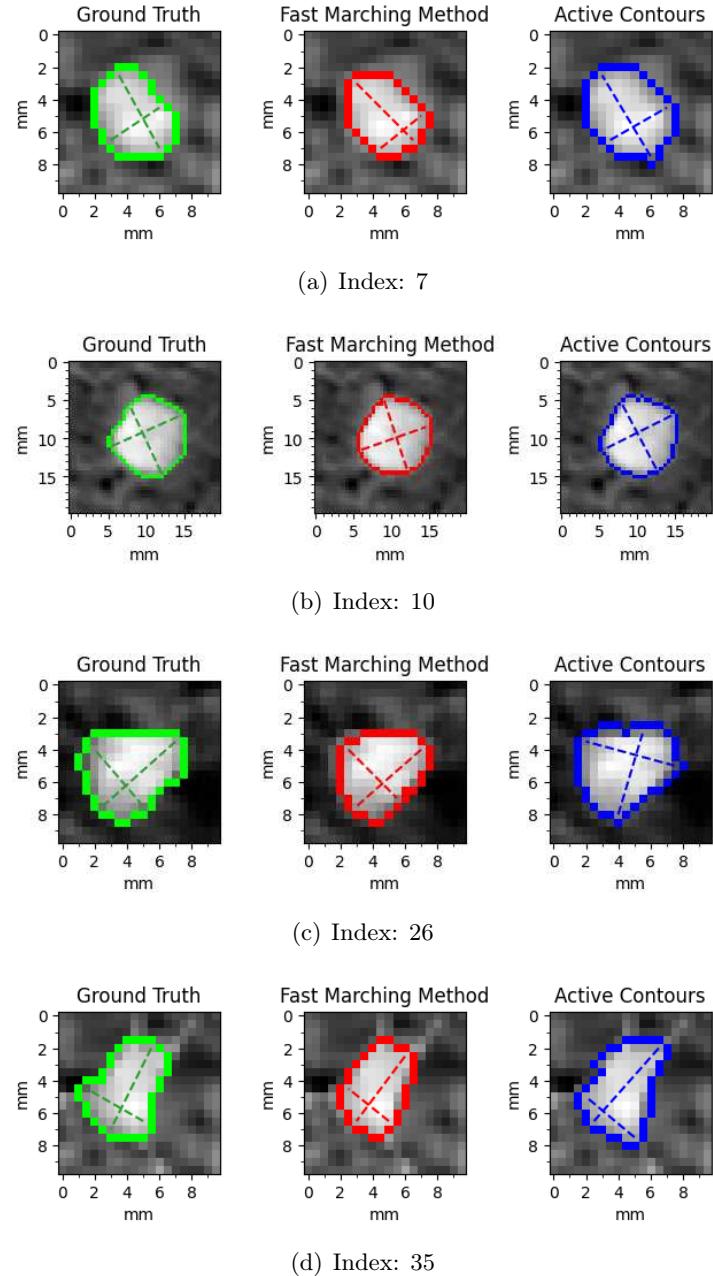


Figure 3.45: AC better than FMM ( $\text{IoU} > 0.5$ )

### 3.3.3.2 Irregular

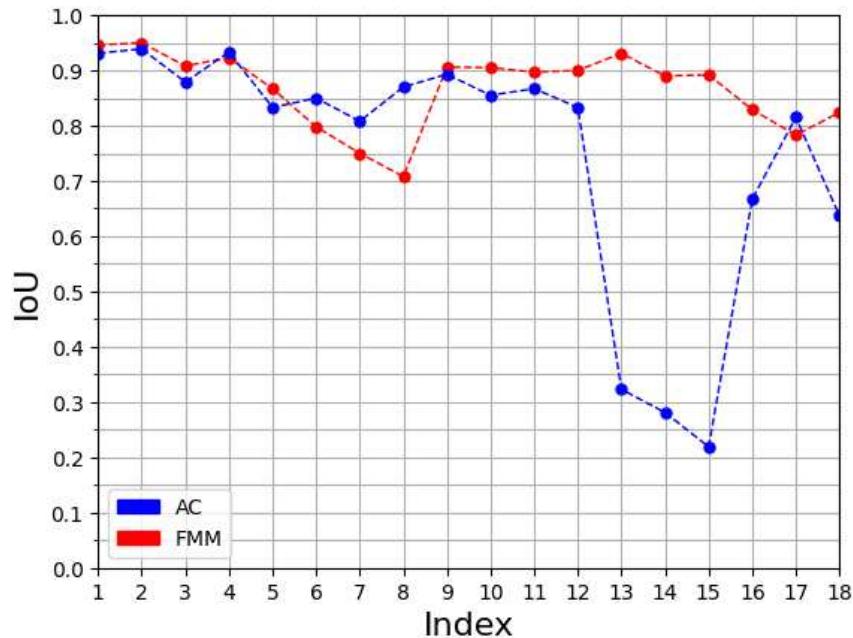


Figure 3.46: Irregular - Intersection over Union (1/2)

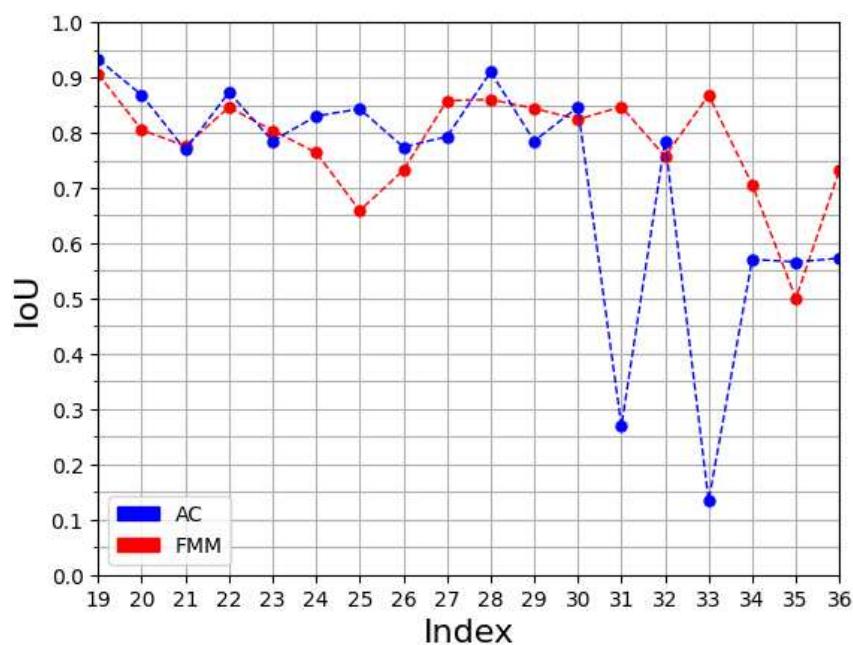


Figure 3.47: Irregular - Intersection over Union (2/2)

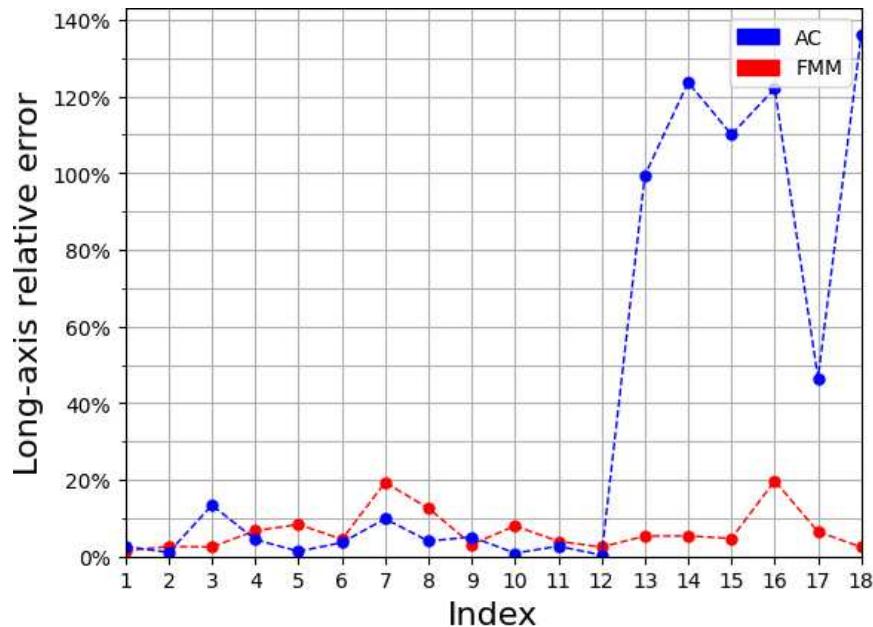


Figure 3.48: Irregular - Long-axis relative error (1/2)

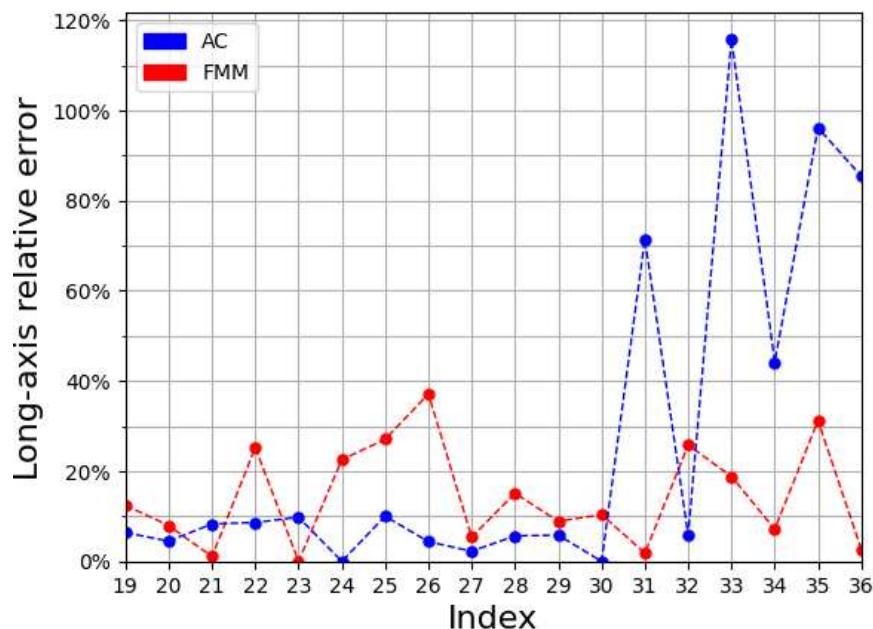


Figure 3.49: Irregular - Long-axis relative error (2/2)

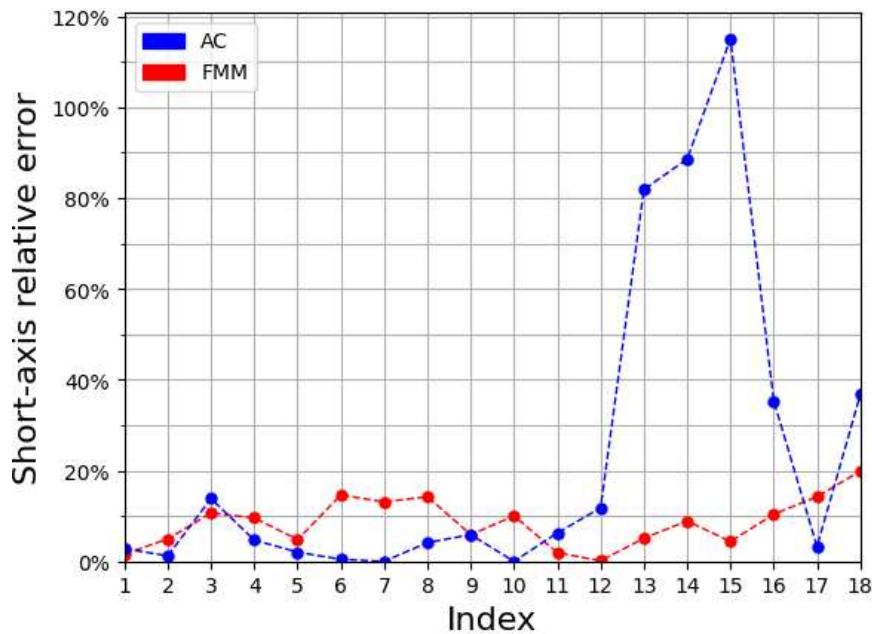


Figure 3.50: Irregular - Short-axis relative error (1/2)

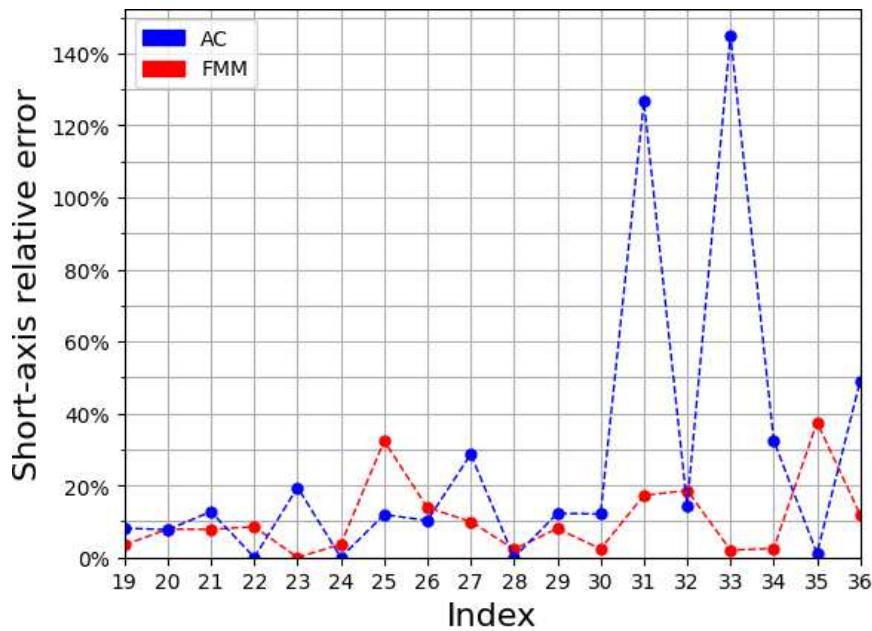


Figure 3.51: Irregular - Short-axis relative error (2/2)

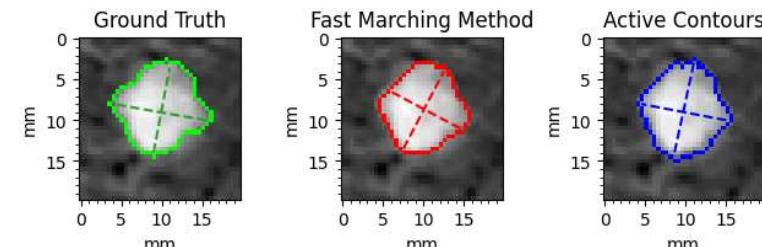
To summarize the results and make a comparison between the two methods, for each metric the mean and standard deviation have been calculated and are shown in table 3.6. The standard deviation is high for both algorithms, especially for AC due to segmentation failure in juxtapleural cases.

	IoU	Long-axis err.[%]	Short-axis err.[%]
FMM	$0.824 \pm 0.091$	$10.6 \pm 9.51$	$9.56 \pm 8.09$
AC	$0.731 \pm 0.219$	$32.5 \pm 44.9$	$25.2 \pm 37.6$

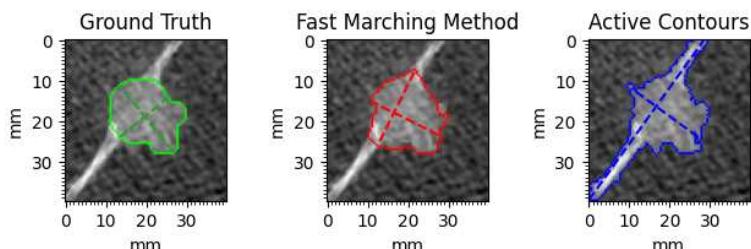
Table 3.6: Irregular objective evaluation results summary

As in the previous case, the fast marching method obtained superior scores for all metrics and overall works better than active contours for segmenting these type of nodules. Similarly to round nodules, irregular nodules are also solid and are well suited for this segmentation method.

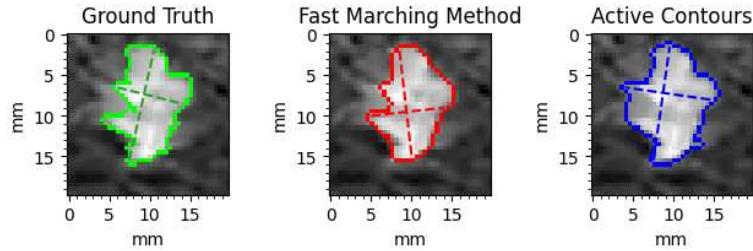
When it come to juxtapleural nodules the behaviour is similar as before, all the segmentation instances with IoU less than 0.5 are due to this shortcoming of the AC method. In the rest of the cases the segmentation performs well. In Figure 3.52 are some examples where FMM outperforms AC, again the juxtapavascular cases are better handled by the FMM algorithm (b,d). Due to the irregular nature of the border, long-axis and short-axis diameter in some cases give different results than IoU, in (a) the IoU score is higher for FMM but also the diameter relative errors. In (c) the long-axis error is higher for FMM, but the short-axis error is higher for AC.



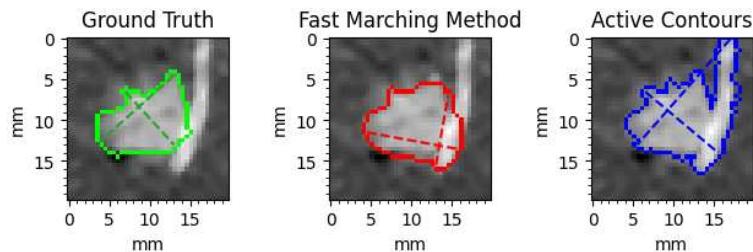
(a) Index: 10



(b) Index: 18



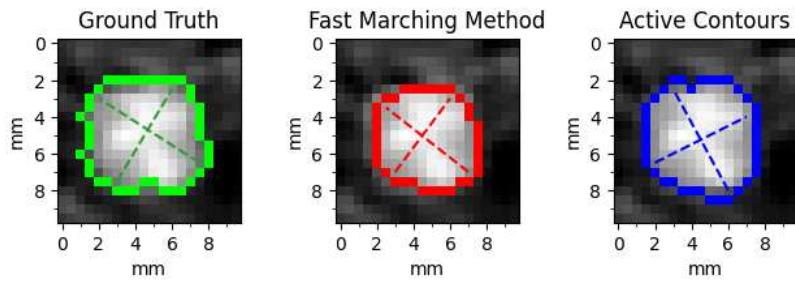
(c) Index: 27



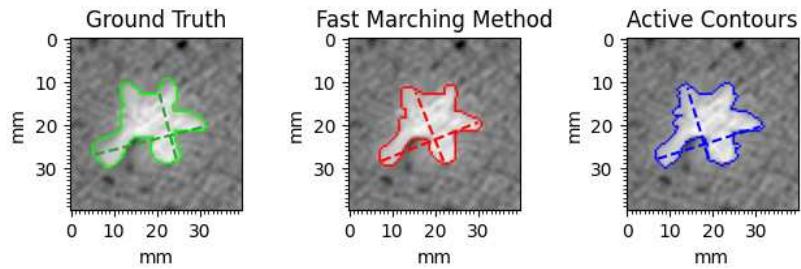
(d) Index: 34

Figure 3.52: FMM better than AC ( $\text{IoU} > 0.5$ )

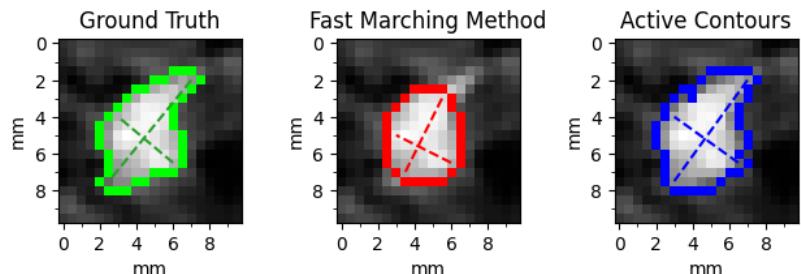
In Figure 3.53 are some cases where AC outperforms FMM. In (b,c) the irregular border is better captured by the AC algorithm, which can handle sharp edges better than FMM, since FMM is region-based and some irregularities are too small to constitute a region and get ignored. In (d) AC achieves a better IoU score, although it also includes the blood vessels, consequently the long-axis error is smaller in the FMM segmentation.



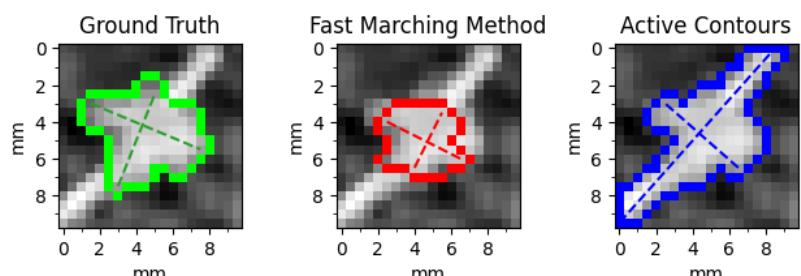
(a) Index: 8



(b) Index: 20



(c) Index: 26



(d) Index: 35

Figure 3.53: AC better than FMM ( $\text{IoU} > 0.5$ )

### 3.3.3.3 Semi-transparent

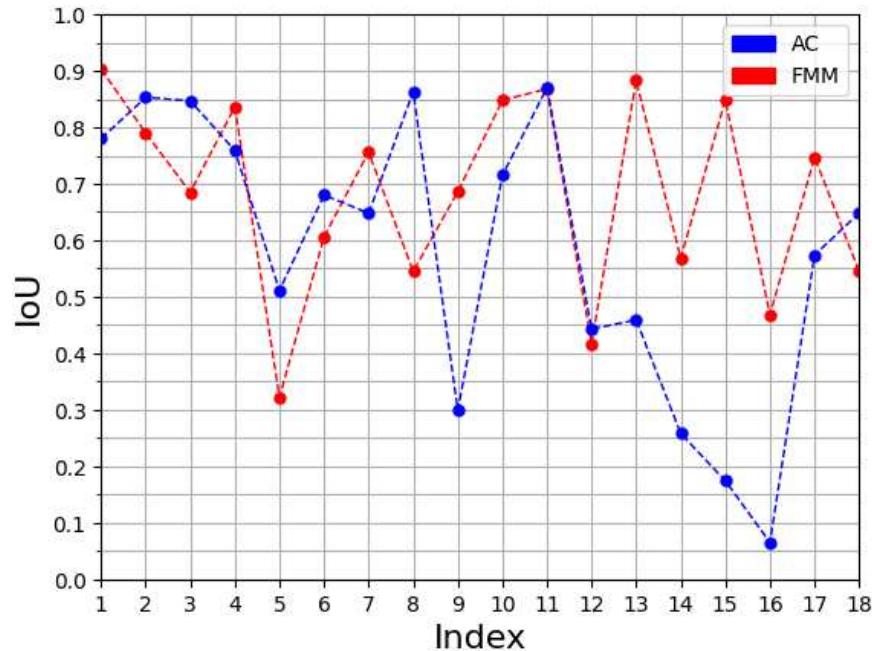


Figure 3.54: Semi-transparent - Intersection over Union

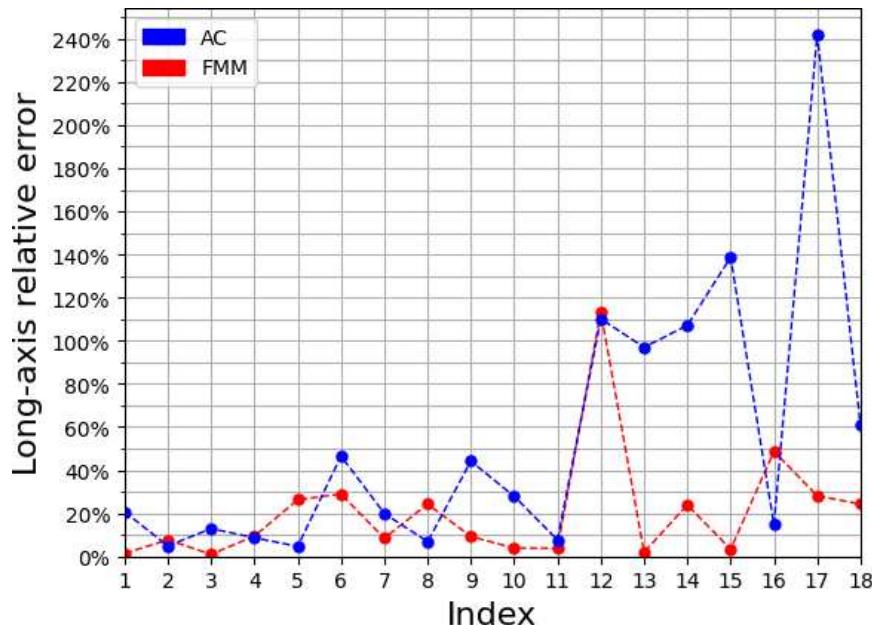


Figure 3.55: Semi-transparent - Long-axis relative error

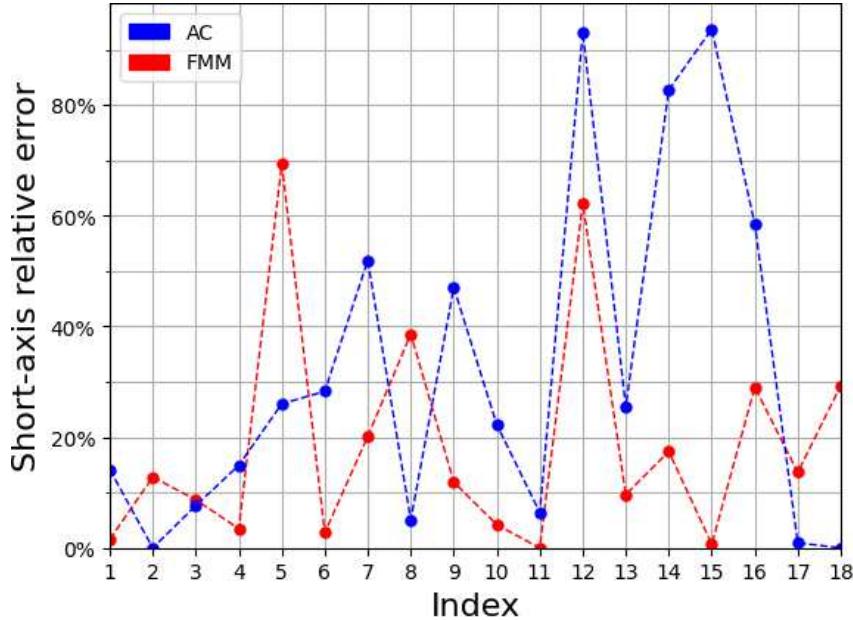


Figure 3.56: Semi-transparent - Short-axis relative error

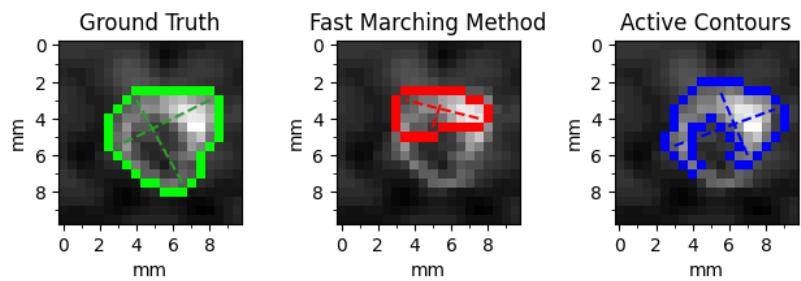
To summarize the results and make a comparison between the two methods, for each metric the mean and standard deviation have been calculated and are shown in table 3.7. For both algorithms the standard deviation is high, because of the frequent cases where segmentation fails.

	IoU	Long-axis err.[%]	Short-axis err.[%]
FMM	$0.684 \pm 0.170$	$20.4 \pm 26.0$	$18.6 \pm 19.8$
AC	$0.580 \pm 0.243$	$54.2 \pm 61.5$	$32.1 \pm 30.9$

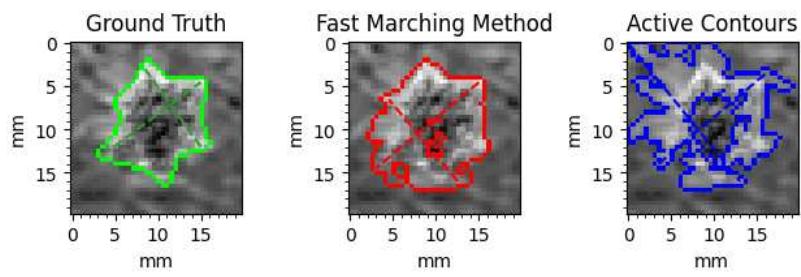
Table 3.7: Semi-transparent objective evaluation results summary

These types of nodules are hard to segment due to the non homogeneous intensity of the nodule tissue and the presence of non-solid parts. Both methods struggle and obtain varied results, although the fast marching method still obtains superior scores on all metrics.

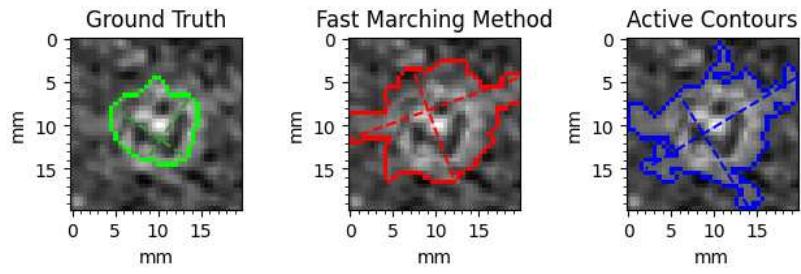
In Figure 3.57 are some examples where the segmentation fails ( IoU less than 0.5 ). In (a) only FMM performs poorly, although AC includes a larger area of the nodule the score is still slightly above 0.5. In (c) both methods fail because they include the surrounding tissue. Same in (f), where FMM includes the blood vessels and surrounding tissue, and AC only segments the vessel ignoring the nodule. In (e) a similar thing happens as for the previous types, where AC fails to segment juxtapleural nodules ( for nodule index 15 the same happens ). In (b,d) only the AC method fails confusing the surrounding tissue as the boundary.



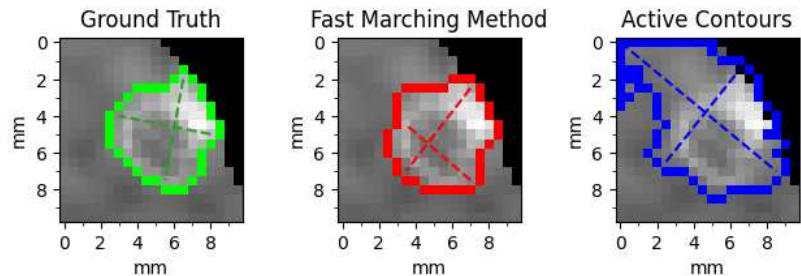
(a) Index: 5



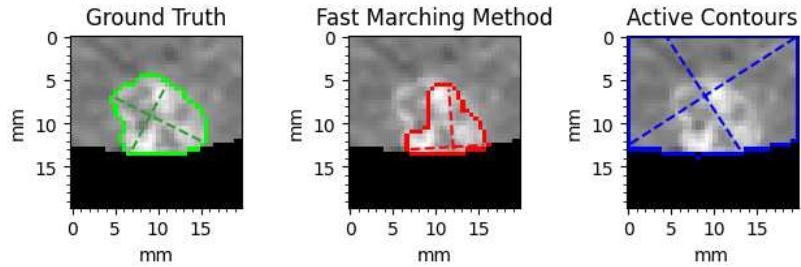
(b) Index: 9



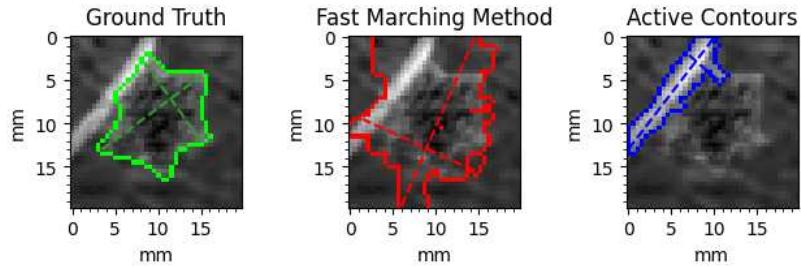
(c) Index: 12



(d) Index: 13



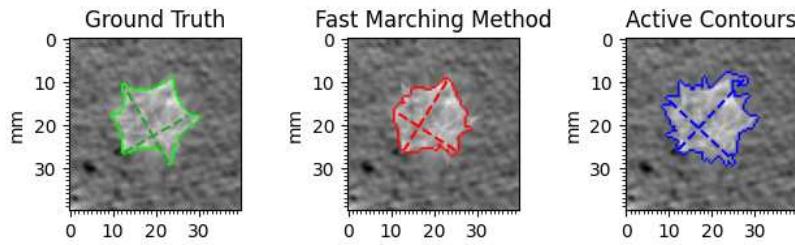
(e) Index: 14



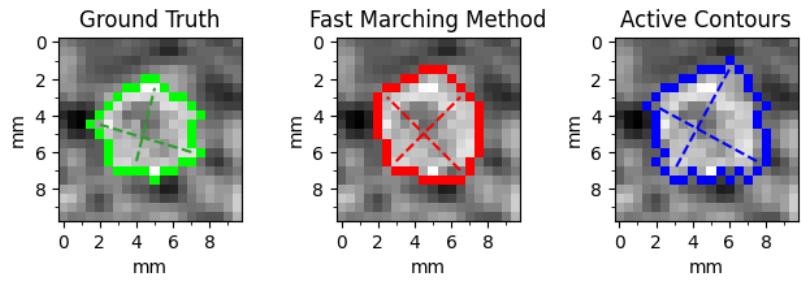
(f) Index: 16

Figure 3.57:  $\text{IoU} < 0.5$

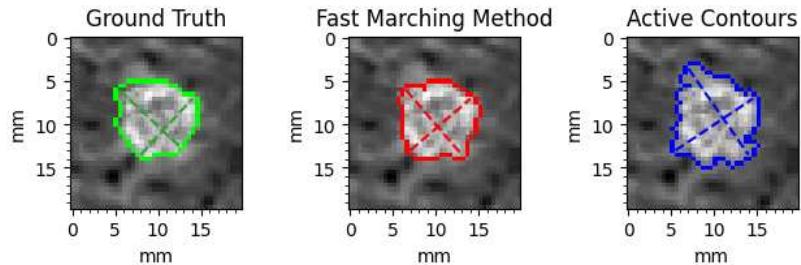
In Figure 3.58 are some examples where FMM outperforms AC, in these cases AC includes more irrelevant surrounding tissue and thus obtains a lower IoU score. In (d) AC segmentation also includes the blood vessel, since it occupies a relatively small area the IoU doesn't suffer much, but this also causes the long-axis measurement to be highly inaccurate.



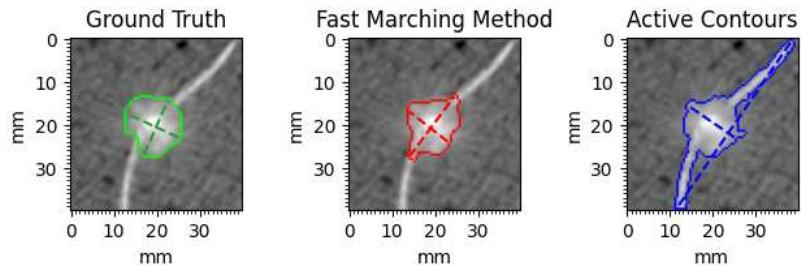
(a) Index: 4



(b) Index: 7



(c) Index: 10



(d) Index: 17

Figure 3.58: FMM better than AC ( $\text{IoU} > 0.5$ )

In Figure 3.59 are some examples where AC outperforms FMM, in (a,b,d) the nodules are part-solid and AC works better in including the non-solid part of the nodule in the segmentation.

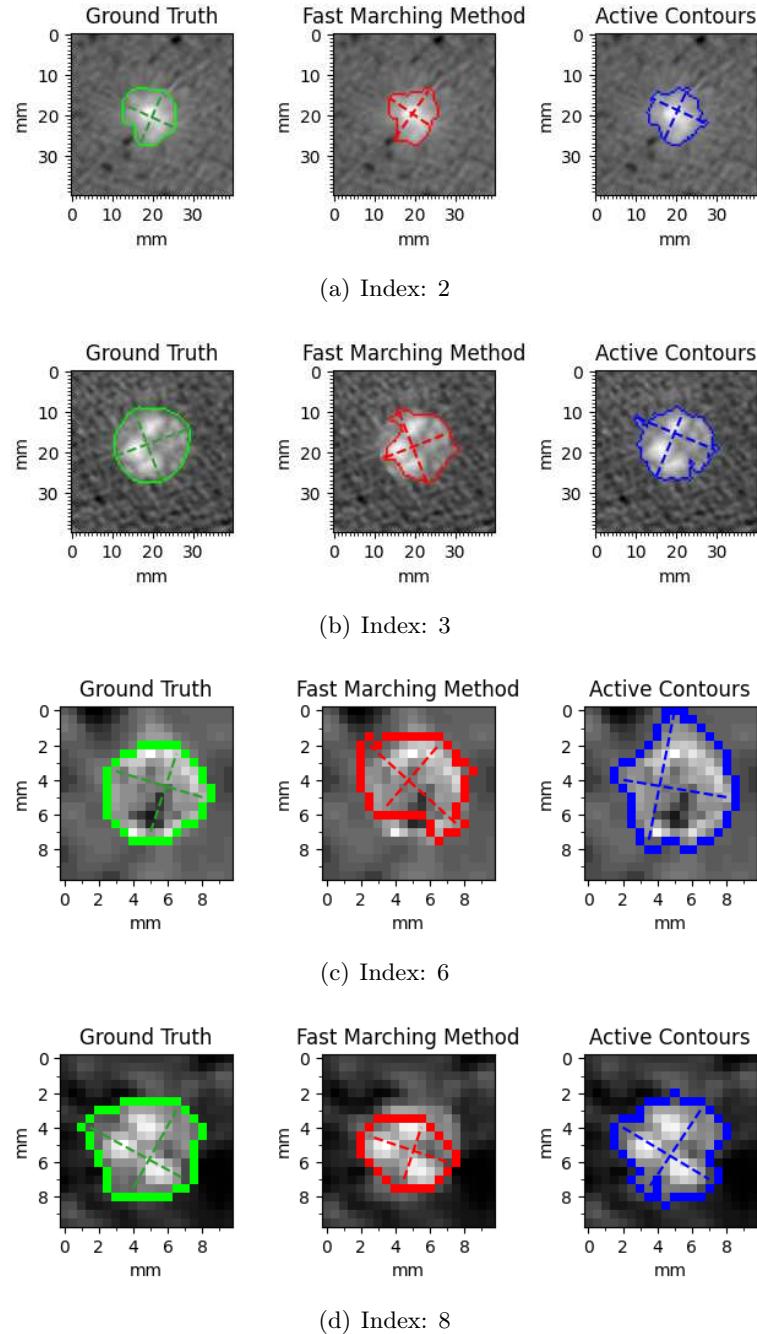


Figure 3.59: AC better than FMM ( $\text{IoU} > 0.5$ )

### 3.3.3.4 Cavitary

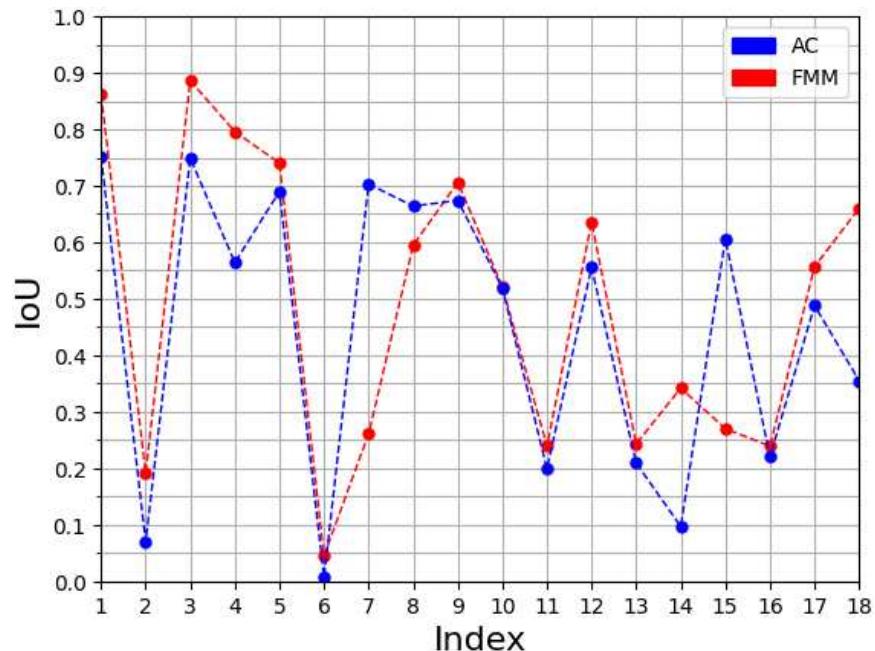


Figure 3.60: Cavitary - Intersection over Union

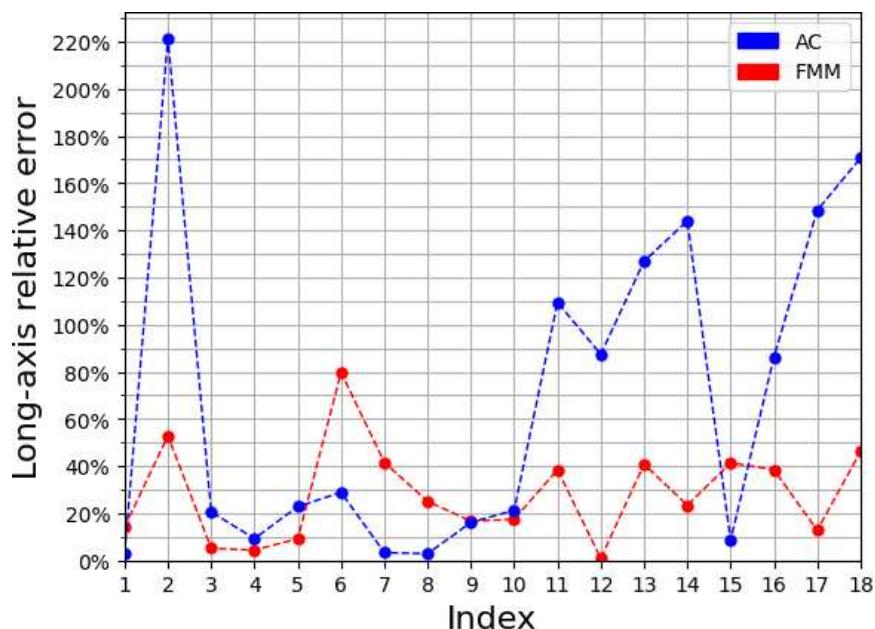


Figure 3.61: Cavitary - Long-axis relative error

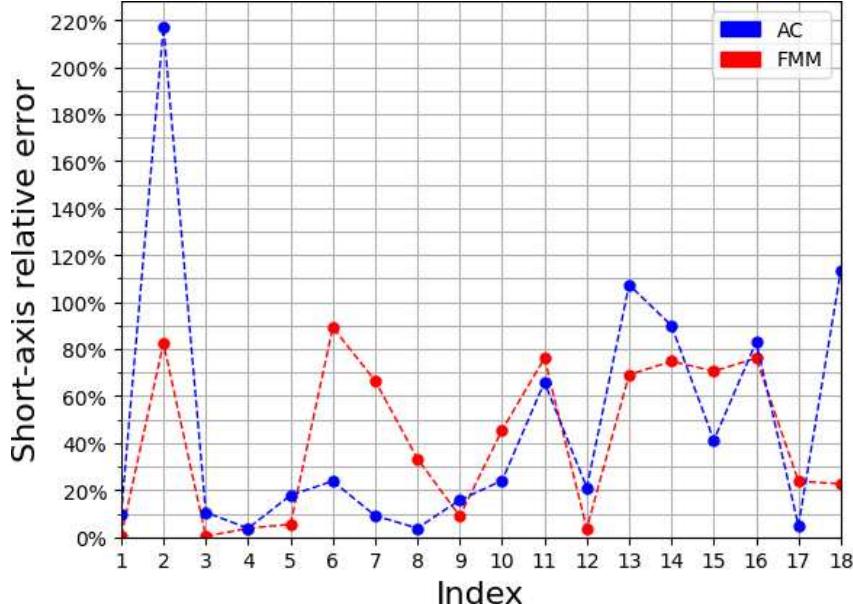


Figure 3.62: Cavitary - Short-axis relative error

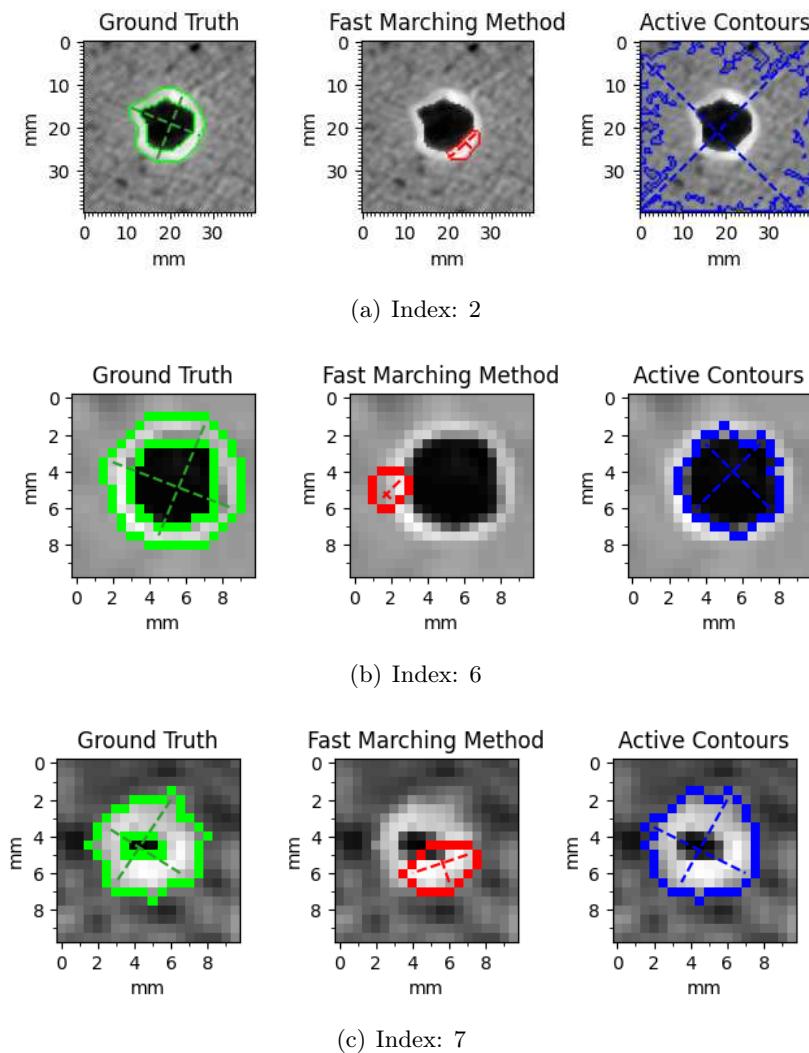
To summarize the results and make a comparison between the two methods, for each metric the mean and standard deviation have been calculated and are shown in table 3.8. For both algorithms the standard deviation is high, because of the frequent cases where segmentation fails.

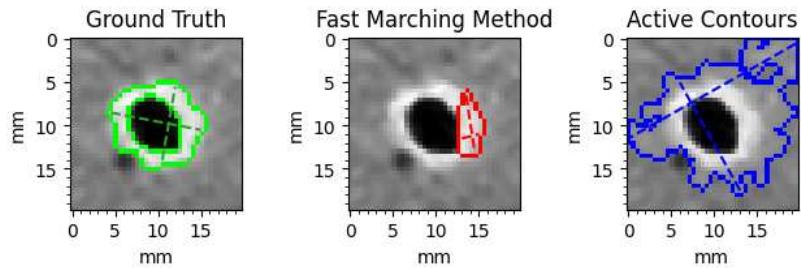
	IoU	Long-axis err. [%]	Short-axis err. [%]
FMM	$0.488 \pm 0.254$	$28.2 \pm 20.0$	$41.8 \pm 32.5$
AC	$0.451 \pm 0.247$	$68.4 \pm 67.7$	$47.9 \pm 54.7$

Table 3.8: Cavitary objective evaluation results summary

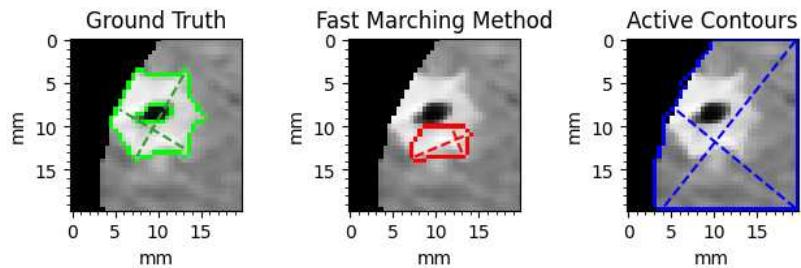
Cavitary nodules have proven to be the toughest targets for segmentation, both methods perform poorly, even though FMM earns moderately higher scores on average. These types of nodules are hard to segment due to the presence of the cavity. In case of FMM, if the nodule tissue around the cavity is too slim it can cause problem with the segmentation since it tends to merge larger clusters. The AC algorithm doesn't take into account the cavity and gives only a single contour boundary or in the worst case confuses the cavity boundary with the nodule. A more accurate segmentation of cavitary nodules with AC could be possible by running the algorithm again on the segmented nodule, thus segmenting the cavity and obtaining a second boundary. But this approach would create new processing costs and difficulties.

In Figure 3.63 are some examples where the segmentation fails ( IoU less than 0.5 ). In (a,b,d,e,f,h) both methods fail, FMM includes too little of the nodule and AC encompasses too much of the surroundings and cavity. In (b) AC fails to give a good segmentation because it only includes the cavity, but the diameter measurements give a decent result because the cavity diameter is similar to the nodule diameter. In cases (e,f) AC fails because of the same issue as with the other juxtapleural nodules. In (c,g) FMM performs worse than AC, because the nodules are small and FMM doesn't manage to capture the tissue around the cavity well. In (i,j) FMM performs better because the nodules are juxtavascular, and AC includes the superfluous vessels in its segmentation.

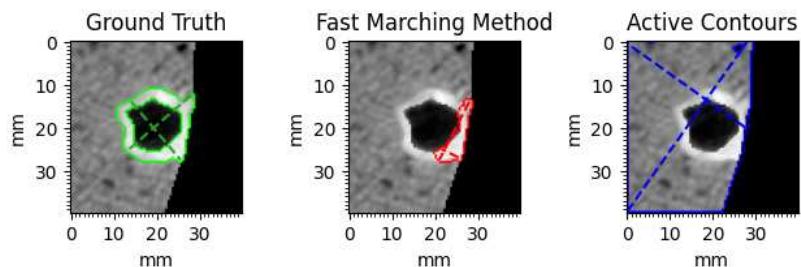




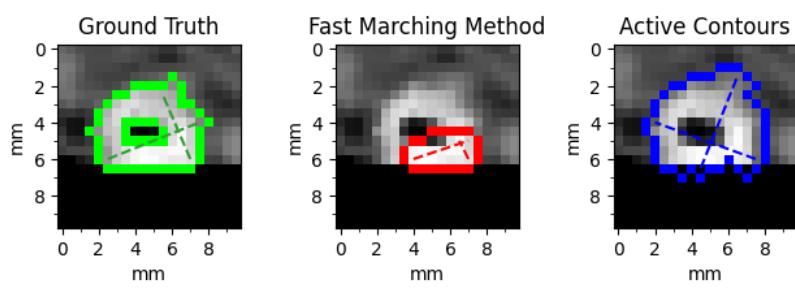
(d) Index: 11



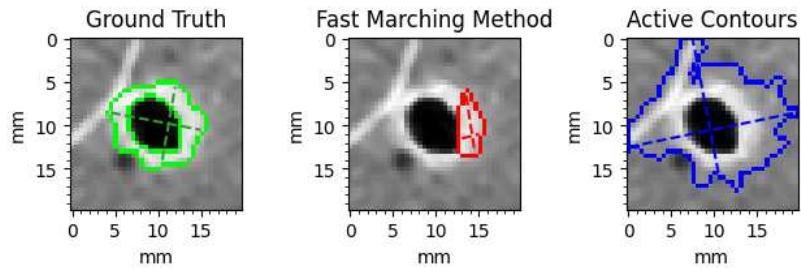
(e) Index: 13



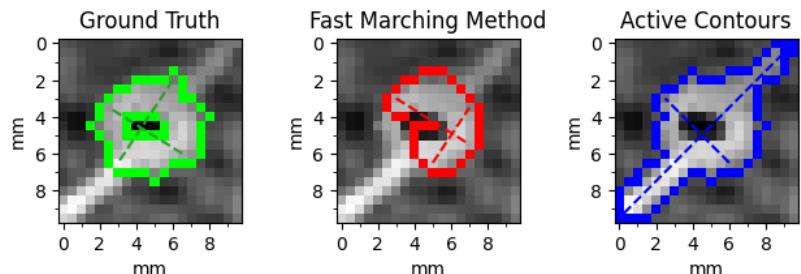
(f) Index: 14



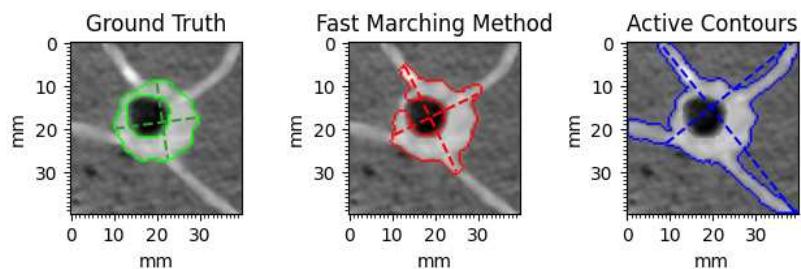
(g) Index: 15



(h) Index: 16



(i) Index: 17



(j) Index: 18

Figure 3.63:  $\text{IoU} < 0.5$

In Figure 3.64 are some examples where FMM outperforms AC. In (a,b,c) FMM scores better on IoU because it excludes the cavity, the diameter measurements are good for both segmentation methods.

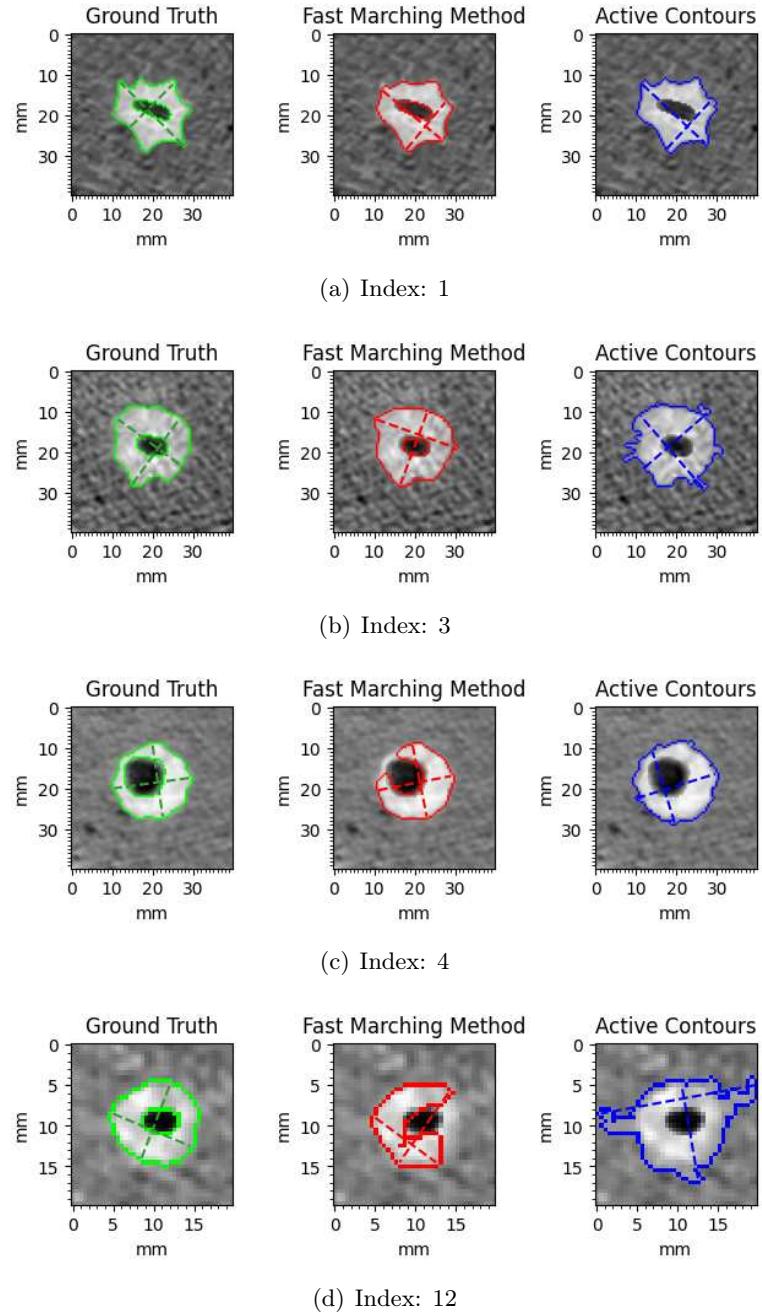


Figure 3.64: FMM better than AC ( $\text{IoU} > 0.5$ )

In Figure 3.65 is the only case where AC performs better than FMM ( for  $\text{IoU} > 0.5$  ). For the AC segmentation, the IoU score is penalized by the inclusion of the cavity, but diameter measures stay relatively close to the ground truth values.

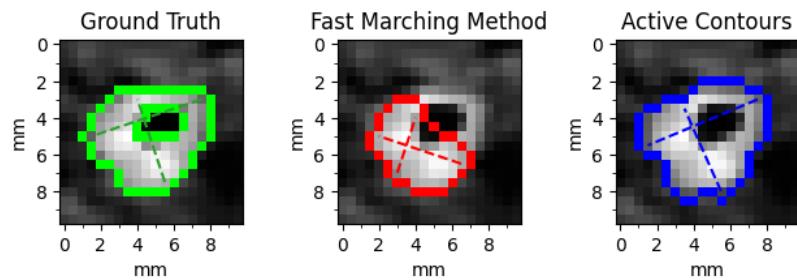


Figure 3.65: AC better than FMM ( $\text{IoU} > 0.5$ ), Index: 8

### 3.4 Discussion and comparison

The two different evaluations have produced sufficient results to gain some insight on how accurately the fast marching method can segment lung nodules. Furthermore, the fact that both datasets produce similar outcomes with completely unrelated evaluation criteria implies that the nodules from the modified phantom dataset are similar to real nodules, and successfully emulate their most relevant features. Overall the fast marching method performed better with the objective evaluation and worse on the subjective evaluation. This phenomenon can be attributed to two factors: the datasets and the evaluation criteria.

The phantom dataset was constructed to be as close as possible to how real nodules appear on a CT image, replicating the most important features that different nodules possess. The only discernible difference that can be attributed to the datasets is the fact that AC segmentation fails on most juxtapleural nodules from the phantom dataset. This is caused by the fact that in the phantom dataset on average the removed lung wall occupies a larger area in the image, and causes AC to ignore the nodule contour by only focusing on the wall.

The other more impactful difference is caused by the evaluation criteria, which prioritise distinct aspects of segmentation.

In the evaluation of round nodules, the radiologist gave higher rating to juxtapavascular nodules which include the blood vessel in the segmentation, whereas the objective method focused more on maintaining the structure of the nodule separate from any surrounding anatomical objects.

When it comes to irregular nodules, the radiologist assigned higher scores to AC in cases where FMM couldn't capture in detail the sharp edges along the nodule border. However, FMM performed better in case of large obstructions, which it can partly ignore, wherein AC includes them in the segmentation. AC has the upper hand in segmenting the boundary in a detailed way, and FMM excels in conserving the overall shape and area.

In the case of semitransparent nodules the fast marching method is more robust and works well in segmenting the solid part; for the non-solid part it is more conservative, and sometimes it fails to capture it in its entirety. The AC segmentation is more open to adding the non-solid part but it can fail and include too much surrounding tissue.

In the case of cavitary nodules, an important difference in evaluation is that the objective method takes the cavity into account when calculating IoU, but the radiologist evaluated only based on the outside boundary. Generally FMM obtains better results when the part around the cavity is thicker.

Both evaluations place the segmentation methods on a similar scale, with one prevailing over the other in a case by case basis.

An indirect comparison can also be made between FMM and the most recent segmentation techniques based on Deep Learning. Wang et al. [20] and Cao et al. [21] both proposed lung nodule segmentation methods which obtained competitive results. Their proposed methods have been evaluated on the public LIDC dataset, and produced Dice scores of  $82.15\% \pm 10.76\%$  and  $82.74\% \pm 10.19\%$  respectively. Dice similarity coefficient is a spatial overlap index and a reproducibility validation metric often used for image segmentation [64], for two sets A,B it is defined in equation 3.4.1.

$$D(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (3.4.1)$$

Since Cao et al. obtained a slightly higher score with their DB-ResNet approach, the latter will be considered. The Dice score is not very different in form from IoU, both are equivalent, the conversion from equation 3.4.2 can be applied.

$$IoU = \frac{D}{2 - D} \quad (3.4.2)$$

The IoU score from DB-ResNet is equal to  $0.706 \pm 0.054$ . A comparison with the IoU scores obtained from FMM is shown in table 3.9.

	Intersection over Union
Round Phantom	$0.877 \pm 0.058$
Irregular Phantom	$0.824 \pm 0.091$
DB-ResNet ( LIDC )	$0.706 \pm 0.054$
Semi-transparent Phantom	$0.684 \pm 0.170$
Cavitory Phantom	$0.488 \pm 0.254$

Table 3.9: Comparison between FMM and DB-ResNet

Since the two methods have been tested on different datasets, it is impossible to tell which performs better, but based on these scores they can be considered comparable. FMM performed worse than the DB-ResNet average on semi-transparent and cavitory nodules, but in the LIDC dataset these types of nodules are much rarer than solid types, thus they would have a small impact on the average Dice score achieved by DB-ResNet.

# Conclusions

The segmentation method discussed in this thesis takes a region-based approach to the fast marching method and proves to have good accuracy when segmenting lung nodules. Particularly when it comes to solid nodules, where splitting into regions, clustering and merging give excellent results. Regarding juxtapleural cases, the method works well since it takes the segmentation of the lungs' area of the preprocessing stage into account. In juxtavascular cases the method can effectively isolate the nodule from the surrounding vessels. In nodules that are usually tougher to segment, it gives a good overall performance, with varied degrees of success in edge cases. Semitransparent nodules are handled quite well, except for extreme cases where the non-solid part is hard to differentiate from the adjacent tissue. In cavitary cases the method gives an accurate segmentation in nodules with thick walls, but struggles in some cases where the walls around the cavity cover a small area. The segmentation is robust and can give an accurate segmentation of the nodule in most cases, being able to handle properly juxtapleural and juxtavascular nodules.

Furthermore, the efficacy of using a dataset with modified phantom nodules was tested, reaching similar results as the dataset with real patient data. The evaluation of two separate methods demonstrated to be useful to analyze where the two methods yield dissimilar results. It can be concluded that a radiologist and an automated method based on metrics like IoU and diameter measure, can provide complementary assessments. Both are beneficial for different reasons: the radiologist can give a more detailed evaluation based on complex features that are of interest, wherein the objective metrics can gauge how well an automated size measurement would perform based on the segmentation.

## Appendix A

### Segmentation results - LIDC

Following are all the results obtained from using the LIDC subset. They are organized by category, and for each nodule there are two separate segmentation instances, one for FMM and one for AC.

The nodules are indexed in the same way as in the Evaluation section.

Each nodule is identified by its LIDC case index and diameter. The subjective evaluation results are displayed in a table containing, for both segmentations, the three scores : area preservation (AP), shape preservation (SP) and overall diagnostic quality (ODQ).

For each nodule, the segmentation boundaries obtained by both methods are displayed in a figure with three images, of which the first is the unaltered image of the nodule obtained from the CT scan, and the other two are the boundaries overlapped on the preprocessed image of the nodule.

---

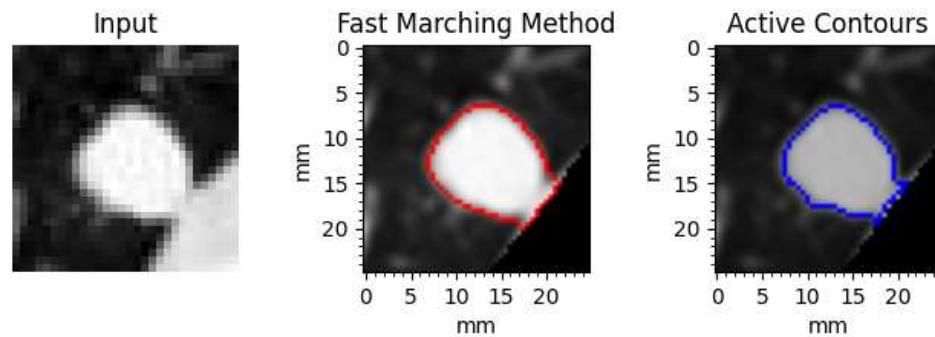
APPENDIX A. SEGMENTATION RESULTS - LIDC

---

### A.1 Round

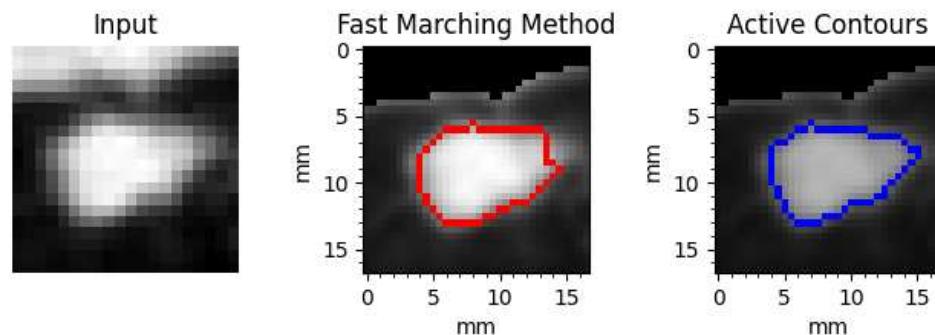
1: LIDC-IDRI-0003-12.82mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



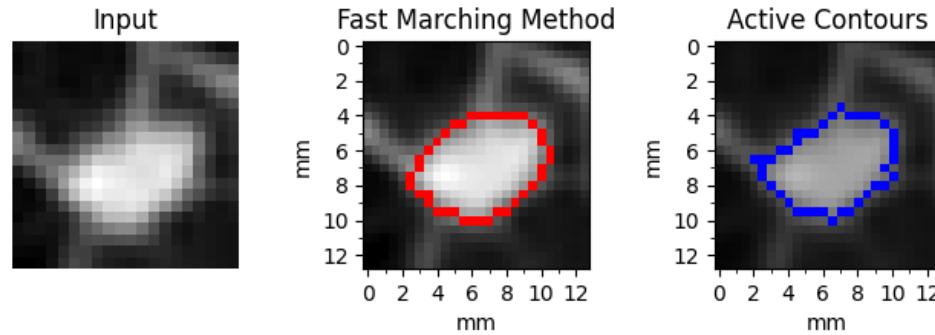
2: LIDC-IDRI-0003-8.79mm

	AP	SP	ODQ
FMM	4	5	5
AC	4	5	5



3: LIDC-IDRI-0006-6.71mm

	AP	SP	ODQ
FMM	4	4	4
AC	4	4	4



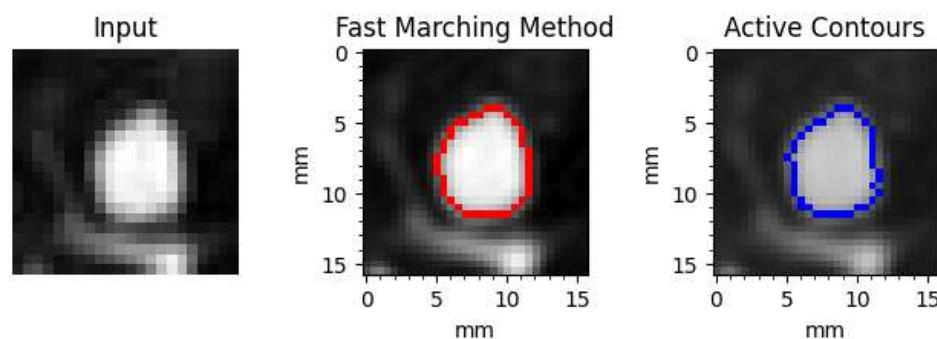
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

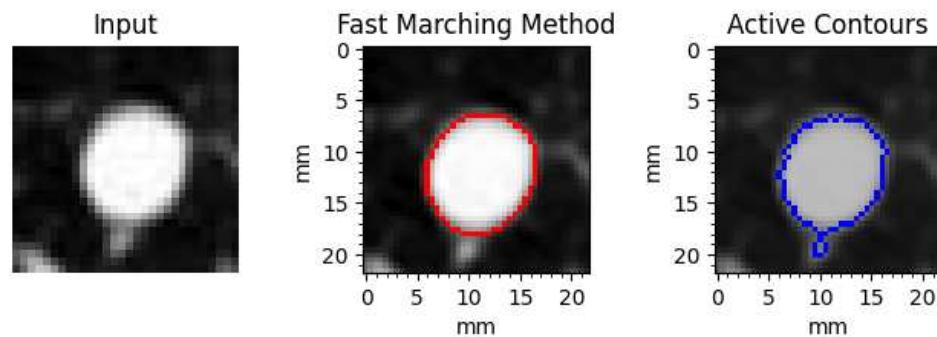
4: LIDC-IDRI-0011-8.44mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



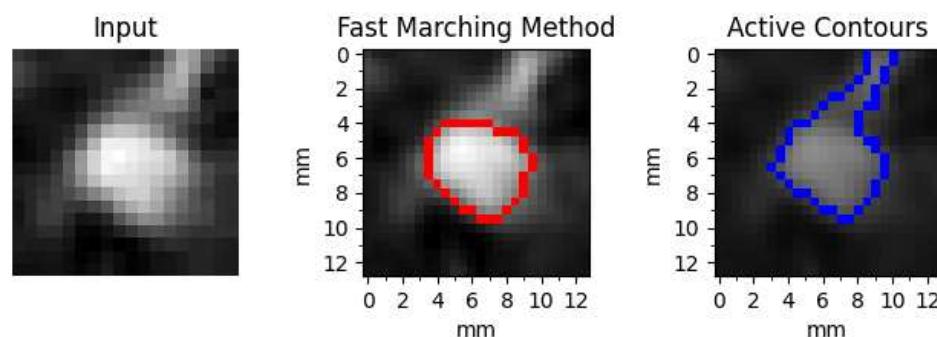
5: LIDC-IDRI-0012-11.41mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



6: LIDC-IDRI-0018-6.5mm

	AP	SP	ODQ
FMM	3	3	3
AC	5	5	5



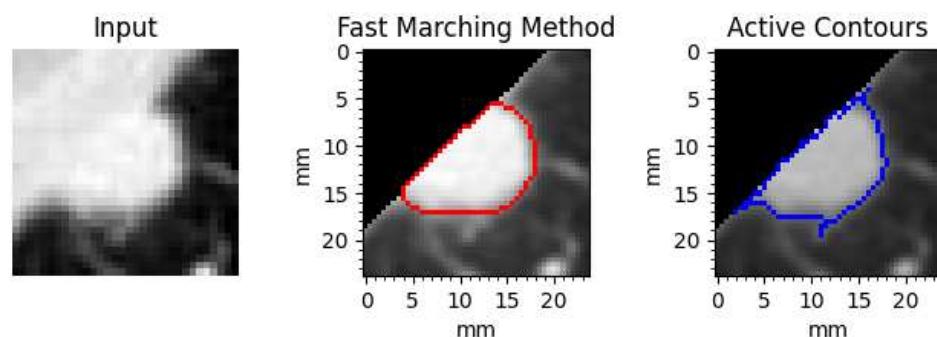
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

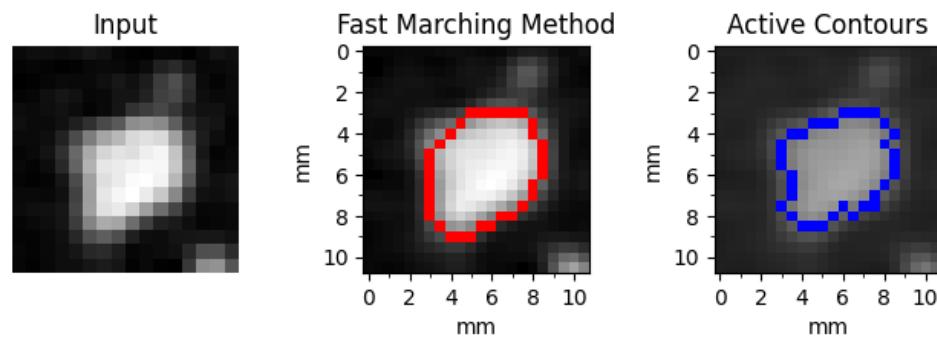
7: LIDC-IDRI-0043-12.46mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



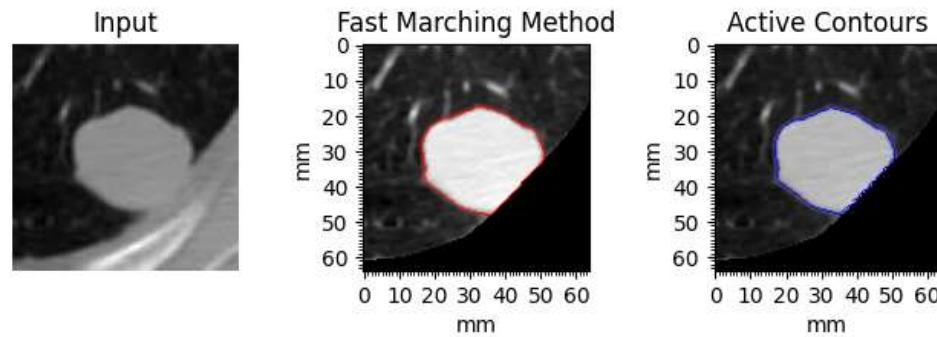
8: LIDC-IDRI-0046-6.0mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



9: LIDC-IDRI-0061-32.27mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



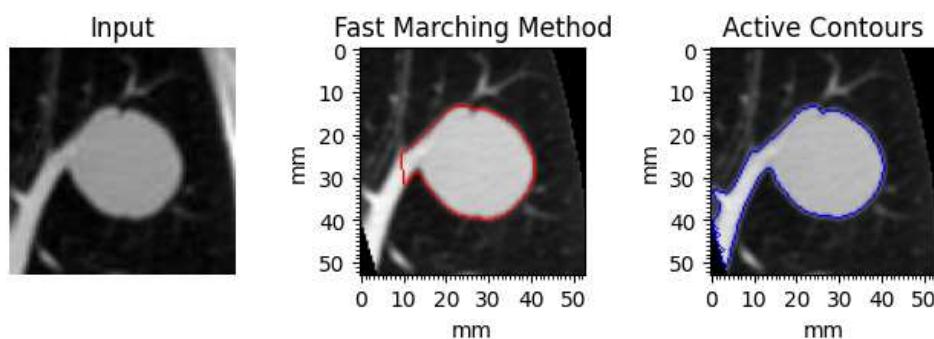
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

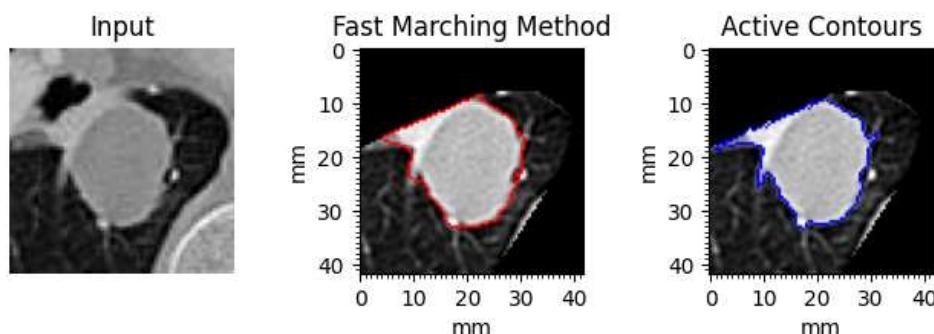
10: LIDC-IDRI-0064-26.84mm

	AP	SP	ODQ
FMM	5	4	5
AC	3	3	3



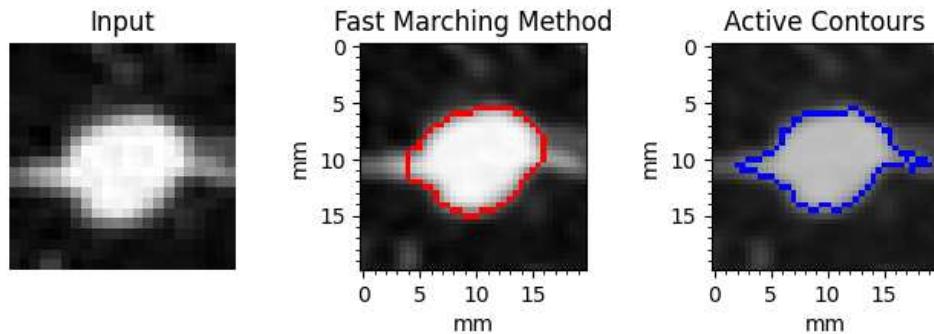
11: LIDC-IDRI-0077-21.15mm

	AP	SP	ODQ
FMM	4	4	4
AC	4	4	5



12: LIDC-IDRI-0081-10.03mm

	AP	SP	ODQ
FMM	3	3	3
AC	5	5	5



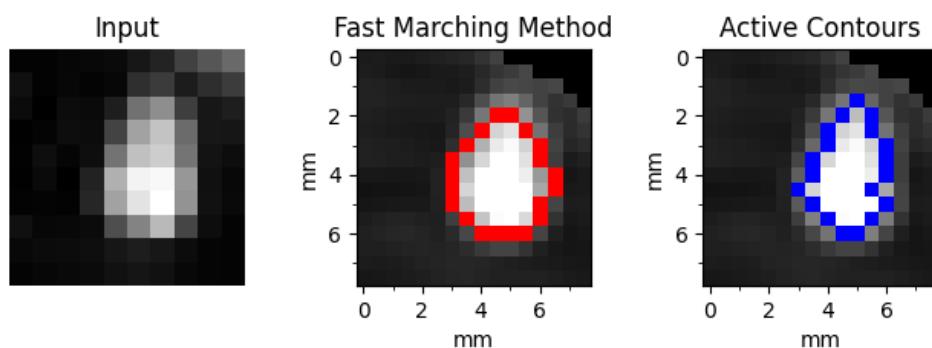
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

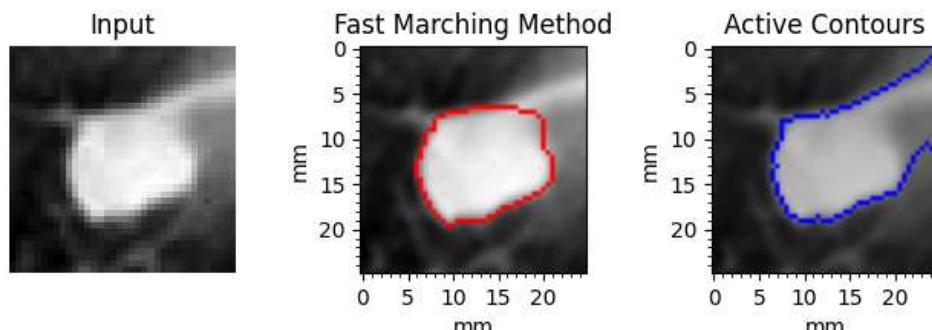
13: LIDC-IDRI-0092-4.02mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



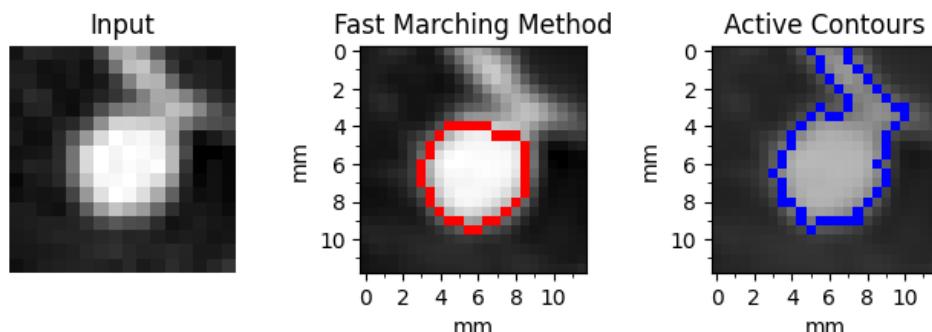
14: LIDC-IDRI-0112-12.69mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



15: LIDC-IDRI-0124-6.48mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



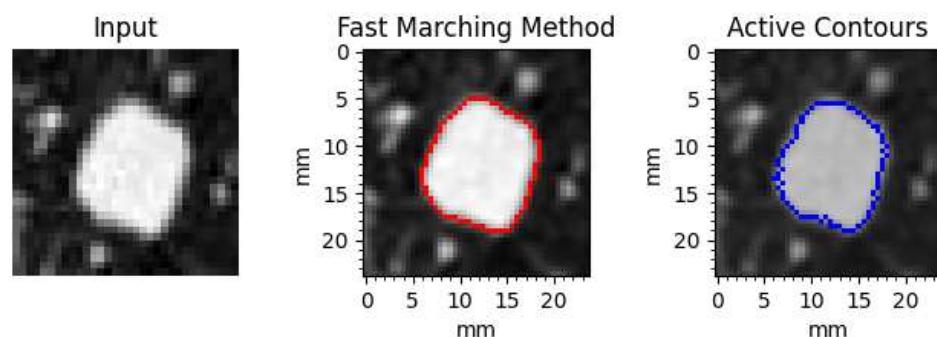
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

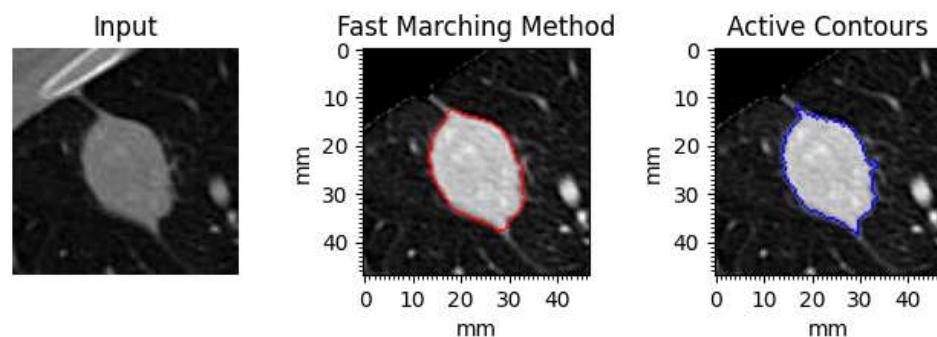
16: LIDC-IDRI-0129-12.38mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



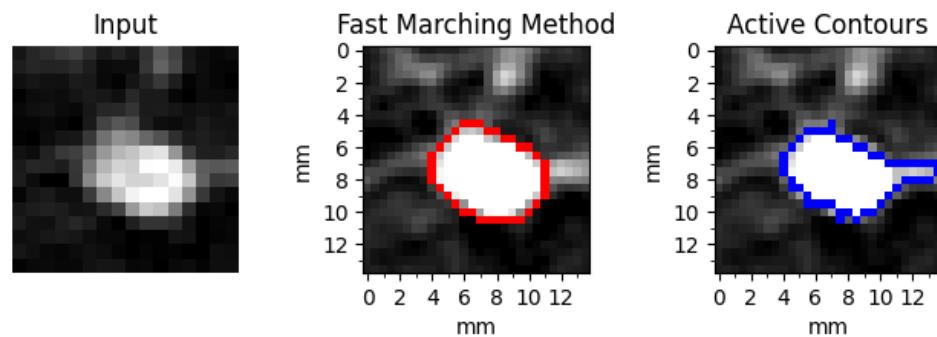
17: LIDC-IDRI-0141-23.93mm

	AP	SP	ODQ
FMM	5	4	5
AC	5	4	5



18: LIDC-IDRI-0160-7.02mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



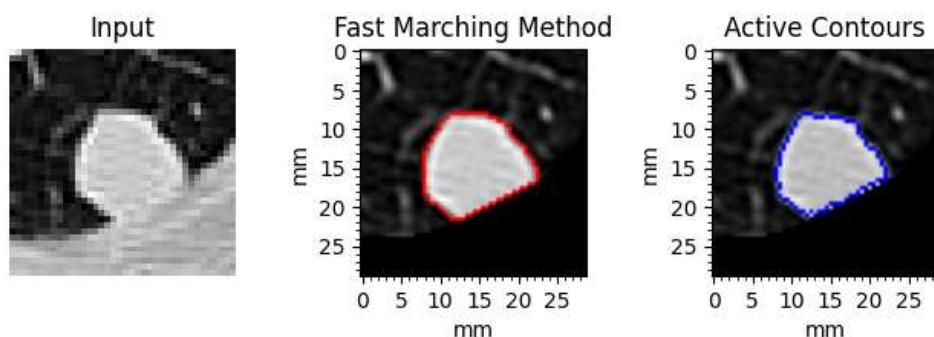
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

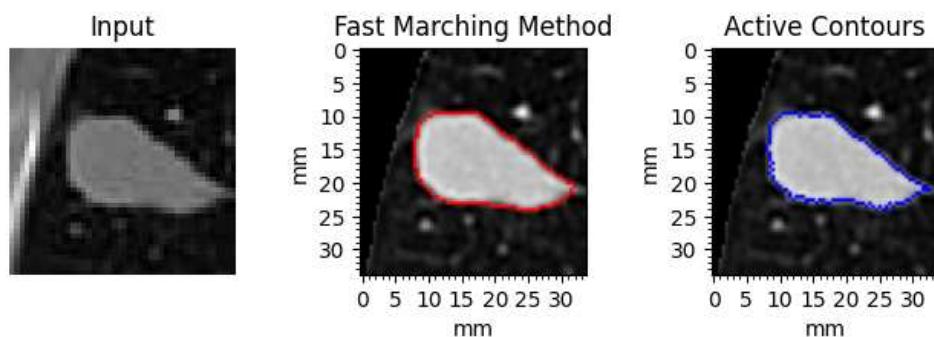
19: LIDC-IDRI-0229-14.84mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



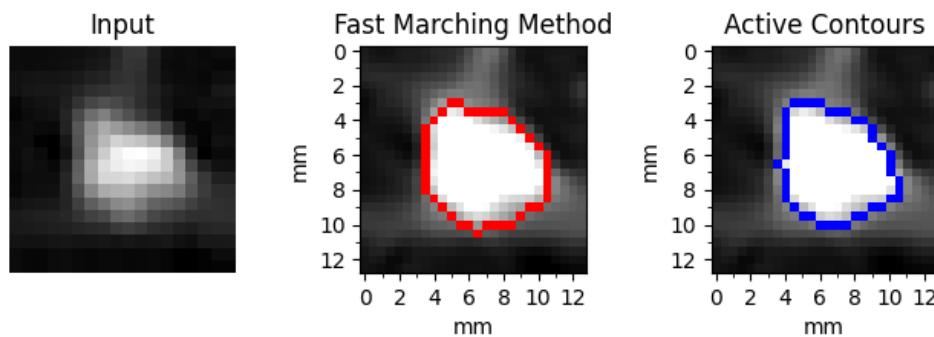
20: LIDC-IDRI-0229-17.4mm

	AP	SP	ODQ
FMM	4	5	5
AC	5	5	5



21: LIDC-IDRI-0427-6.86mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



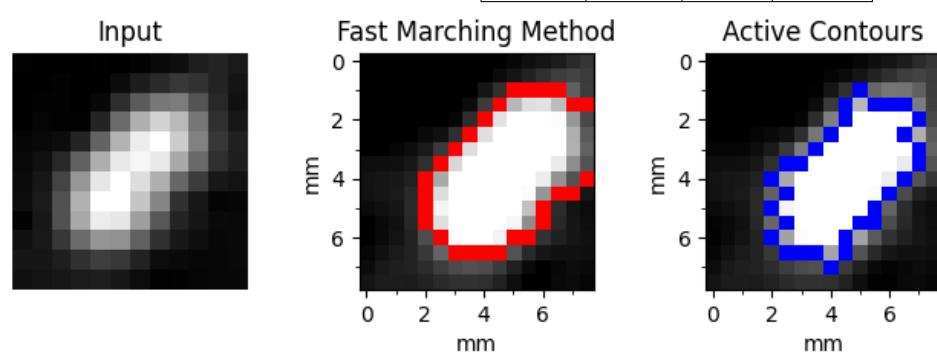
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

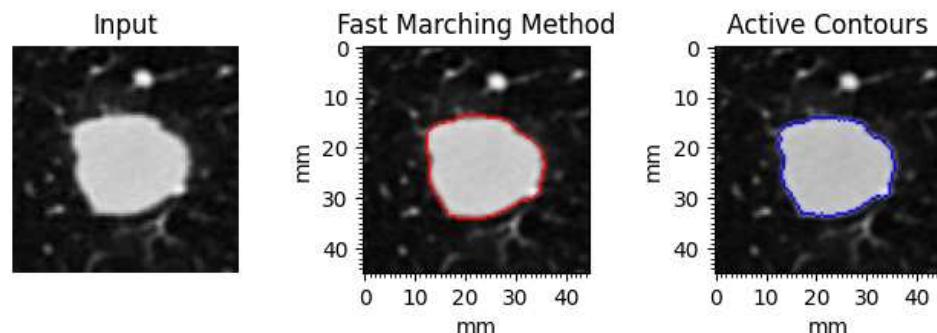
22: LIDC-IDRI-0491-4.07mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



23: LIDC-IDRI-0775-22.78mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



---

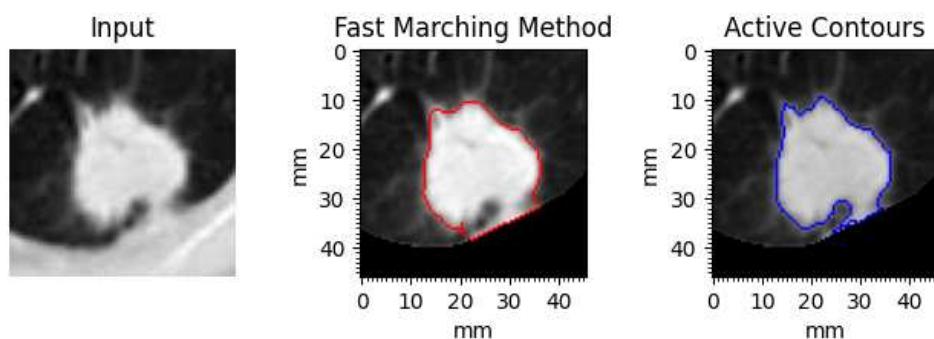
APPENDIX A. SEGMENTATION RESULTS - LIDC

---

## A.2 Irregular

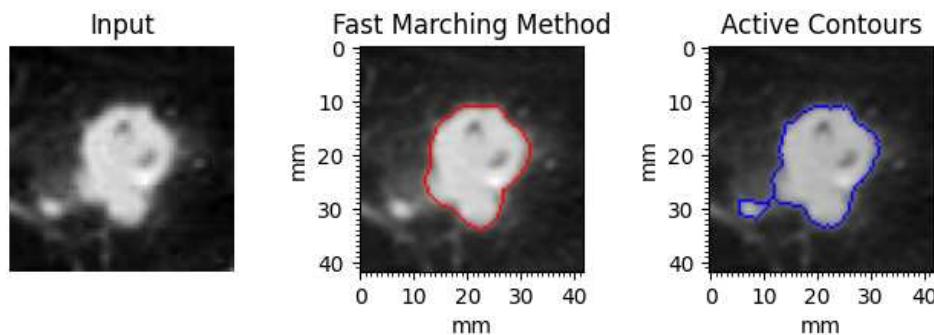
1: LIDC-IDRI-0001-23.35mm

	AP	SP	ODQ
FMM	5	4	5
AC	5	5	5



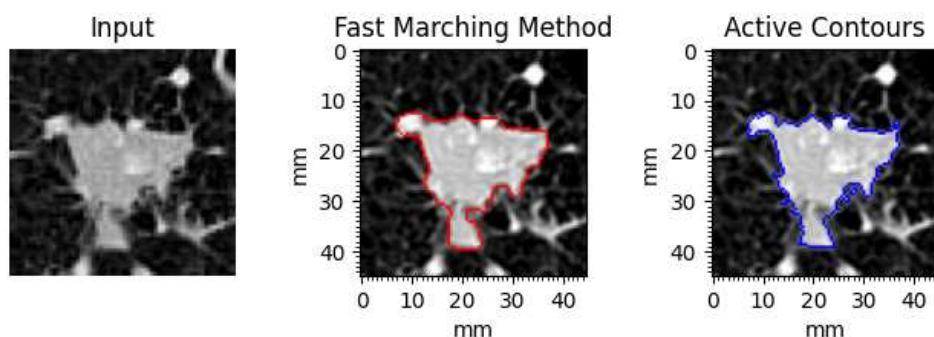
2: LIDC-IDRI-0003-21.45mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



3: LIDC-IDRI-0007-22.88mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



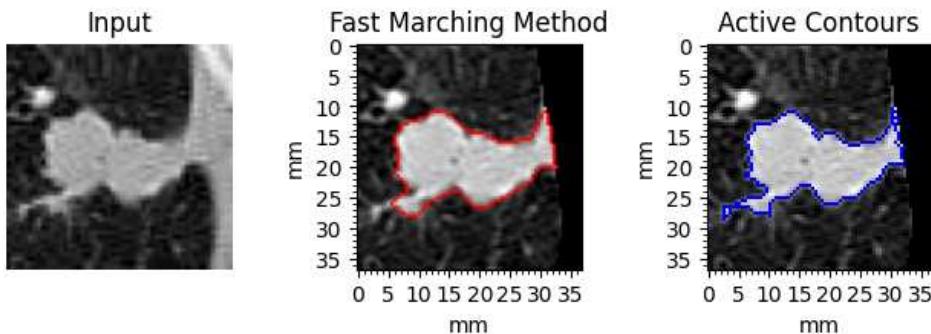
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

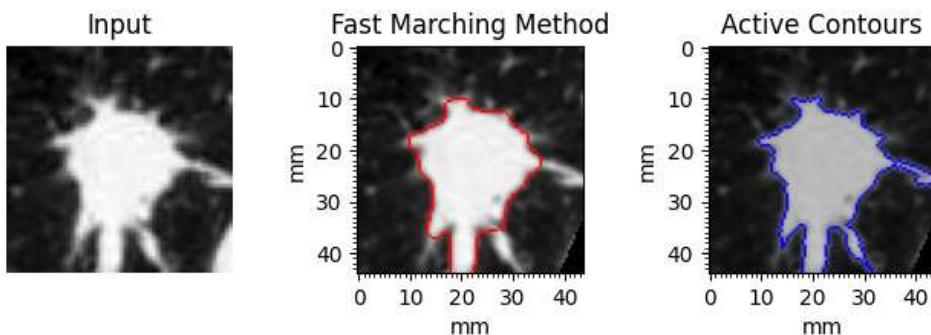
4: LIDC-IDRI-0058-18.87mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



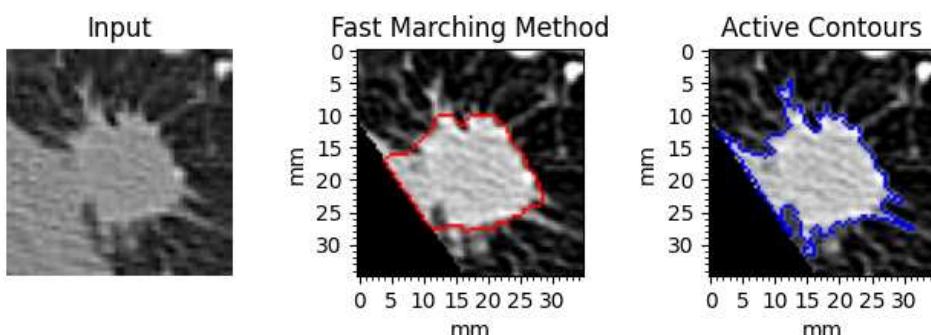
5: LIDC-IDRI-0080-22.24mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



6: LIDC-IDRI-0082-17.72mm

	AP	SP	ODQ
FMM	4	4	5
AC	5	5	5



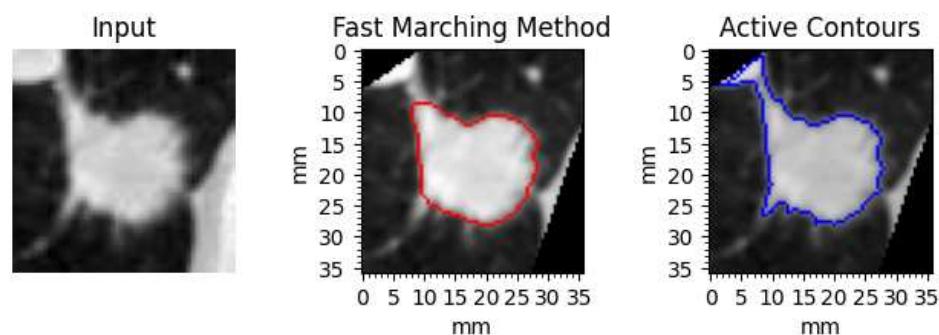
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

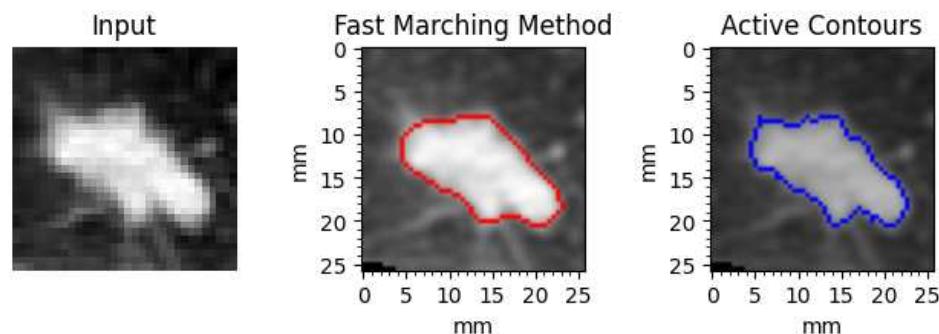
7: LIDC-IDRI-0111-18.02mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



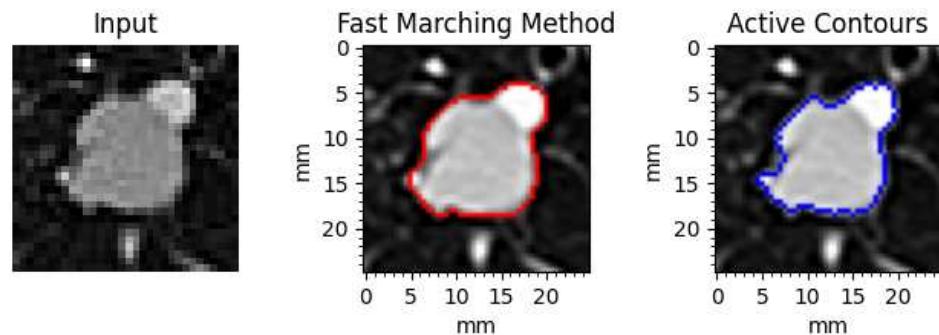
8: LIDC-IDRI-0148-13.03mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



9: LIDC-IDRI-0181-13.0mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



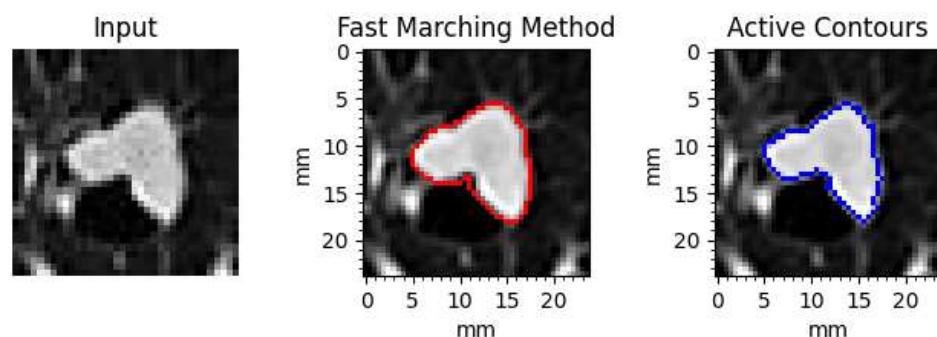
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

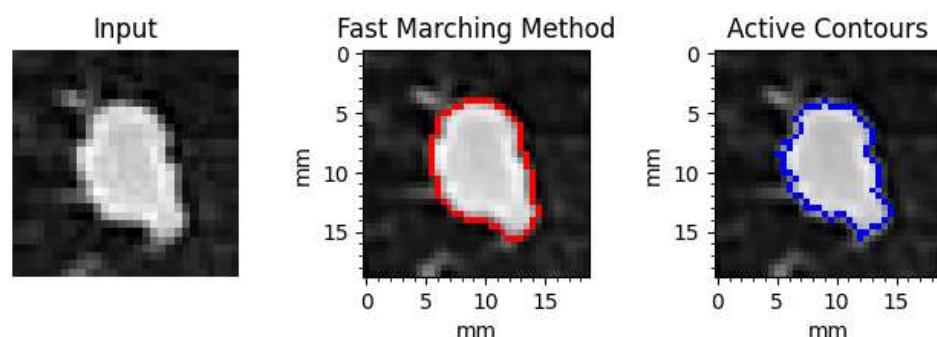
10: LIDC-IDRI-0195-12.03mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



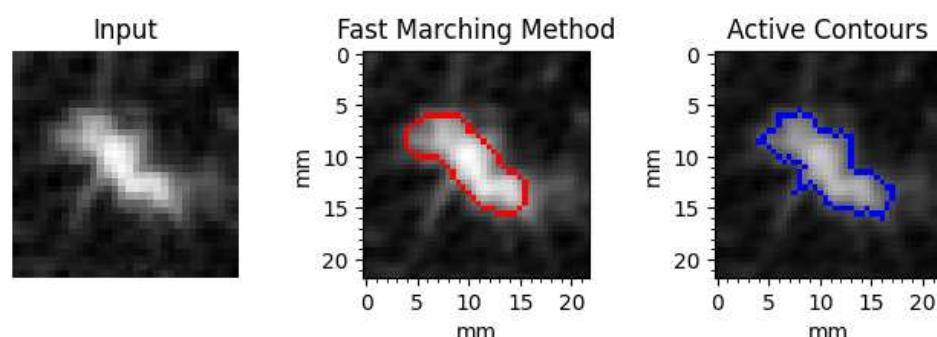
11: LIDC-IDRI-0195-9.88mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



12: LIDC-IDRI-0304-11.3mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	4	5



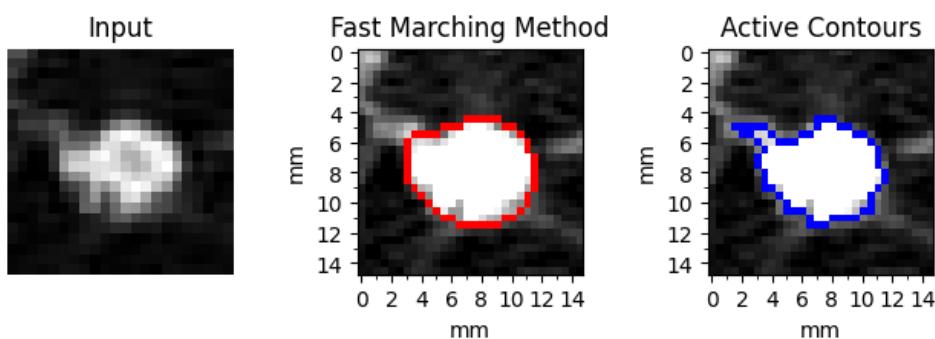
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

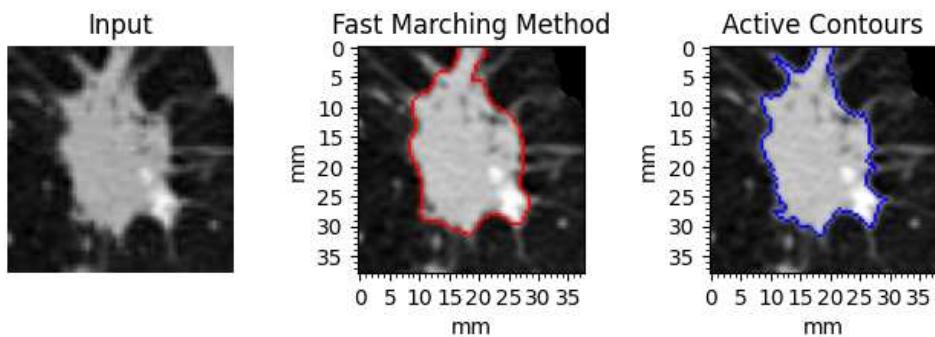
13: LIDC-IDRI-0321-7.92mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



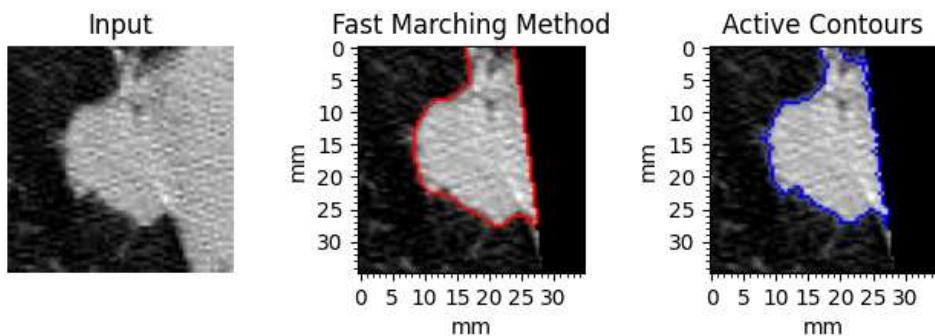
14: LIDC-IDRI-0454-19.31mm

	AP	SP	ODQ
FMM	4	4	5
AC	5	5	5



15: LIDC-IDRI-0466-17.99mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



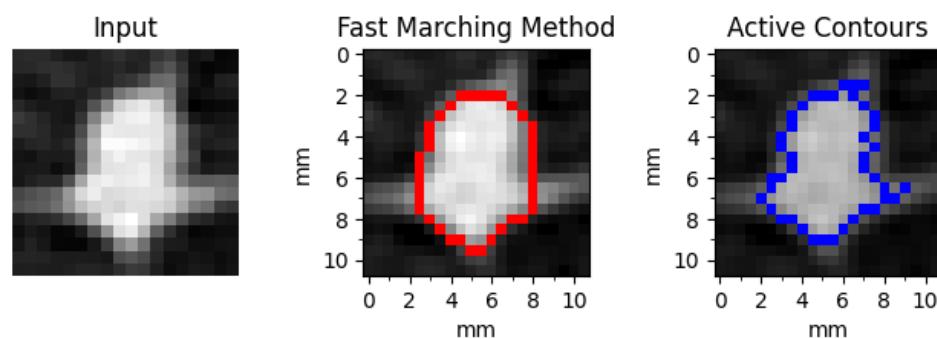
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

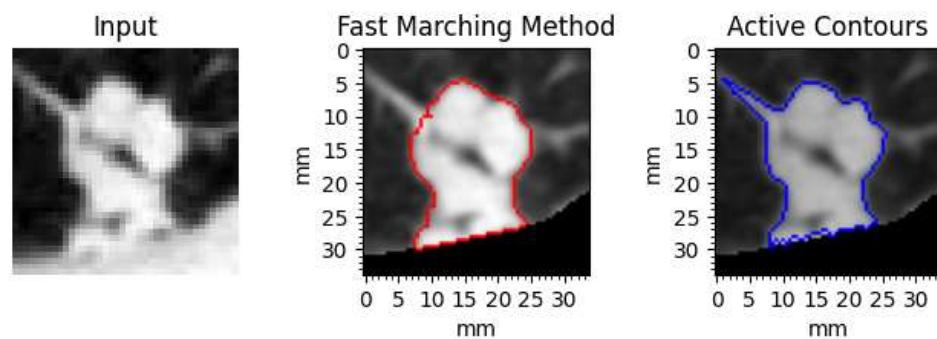
16: LIDC-IDRI-0481-5.64mm

	AP	SP	ODQ
FMM	4	4	4
AC	4	4	5



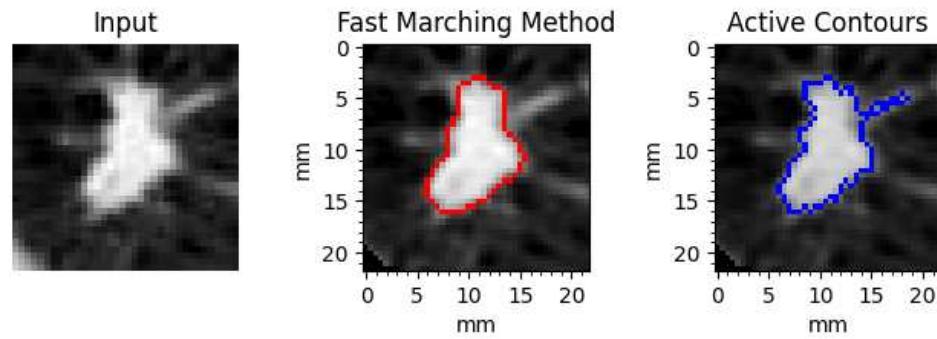
17: LIDC-IDRI-0508-17.15mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



18: LIDC-IDRI-0562-11.1mm

	AP	SP	ODQ
FMM	4	4	4
AC	4	4	5



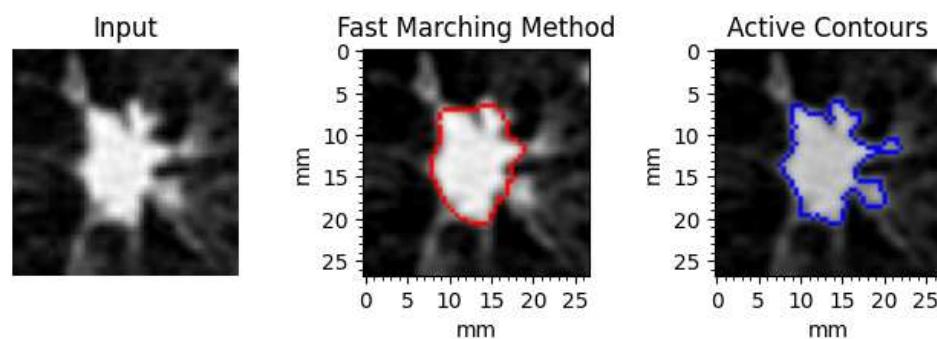
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

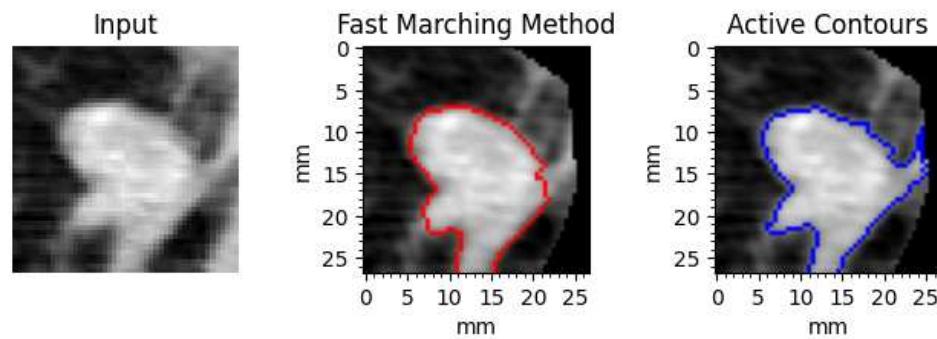
19: LIDC-IDRI-0606-13.68mm

	AP	SP	ODQ
FMM	4	4	3
AC	5	5	5



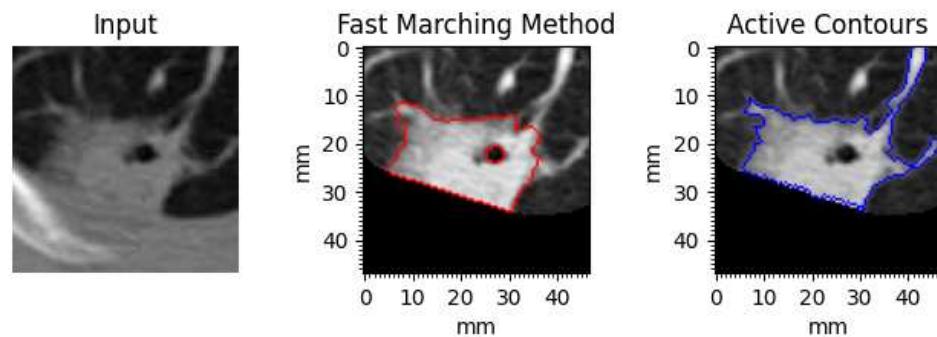
20: LIDC-IDRI-0688-13.64mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



21: LIDC-IDRI-0811-23.57mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



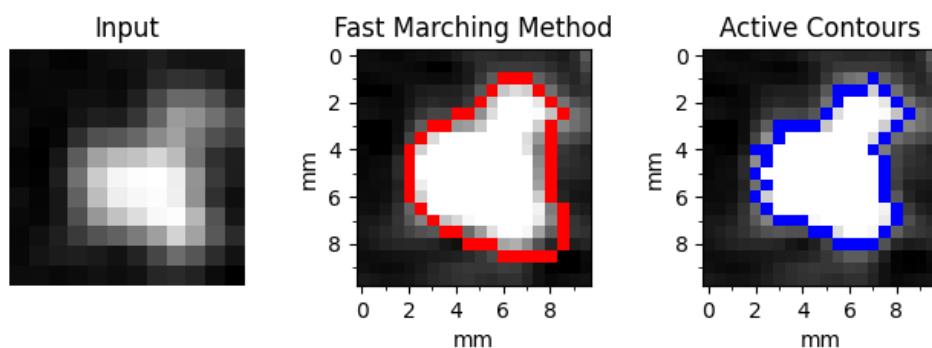
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

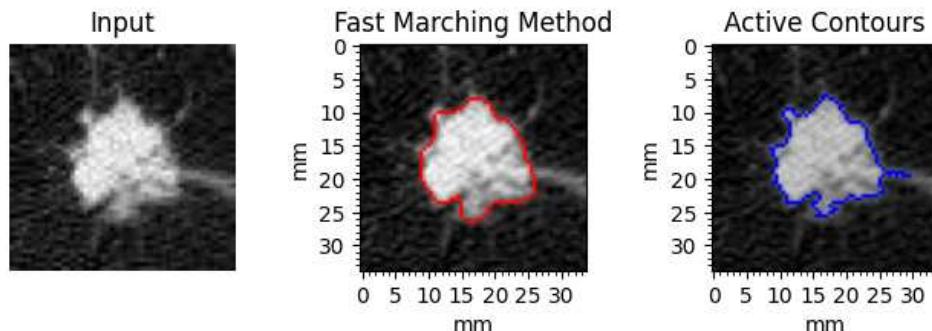
22: LIDC-IDRI-0867-5.25mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



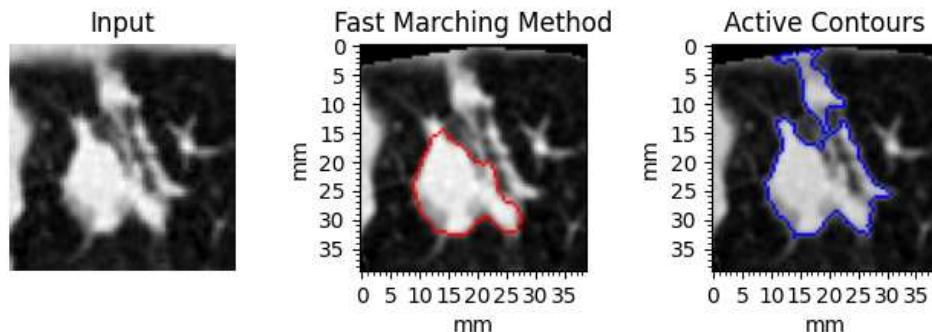
23: LIDC-IDRI-0884-17.41mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



24: LIDC-IDRI-0921-19.85mm

	AP	SP	ODQ
FMM	4	5	5
AC	5	5	5



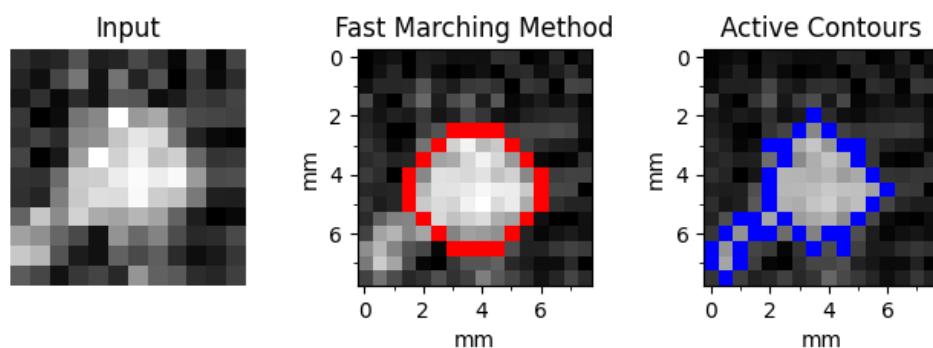
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

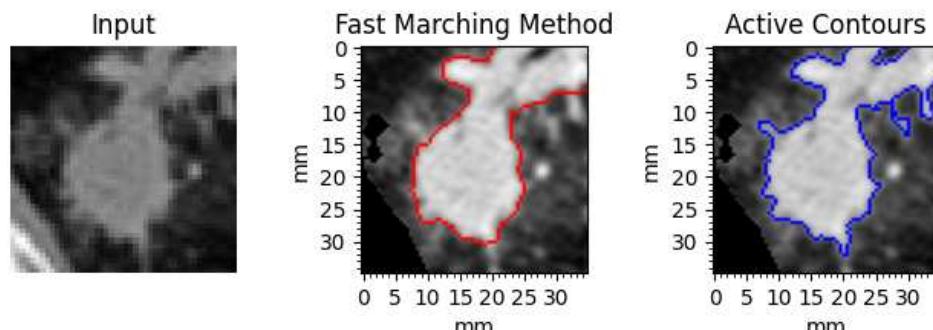
25: LIDC-IDRI-0928-4.03mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



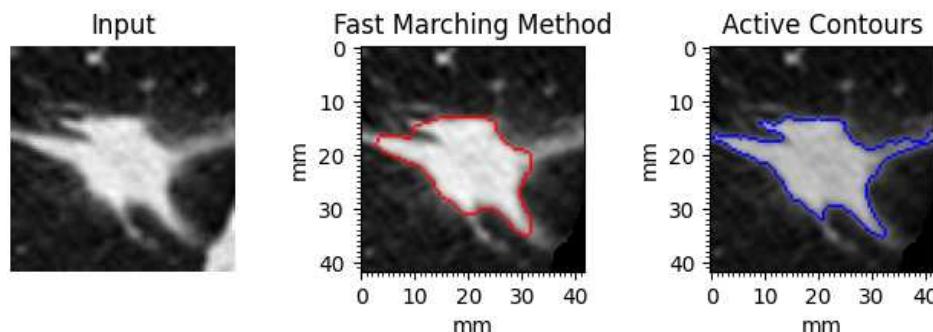
26: LIDC-IDRI-0942-17.69mm

	AP	SP	ODQ
FMM	5	4	5
AC	5	5	5



27: LIDC-IDRI-1007-21.34mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



---

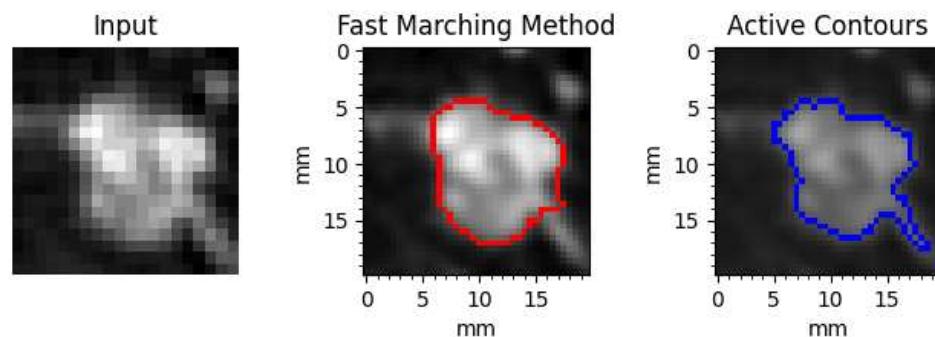
APPENDIX A. SEGMENTATION RESULTS - LIDC

---

### A.3 Semi-transparent

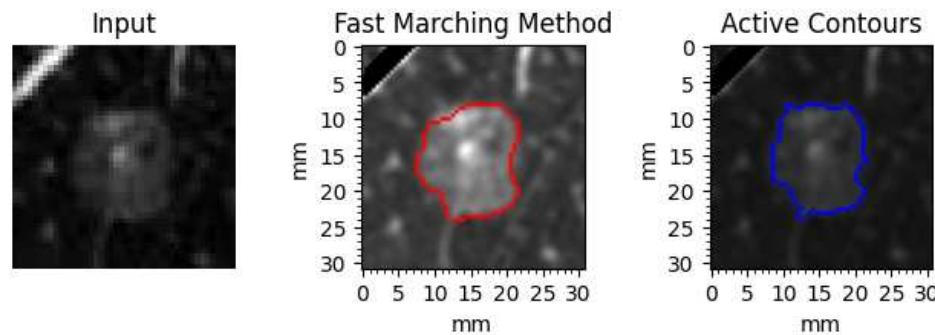
1: LIDC-IDRI-0048-10.06mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



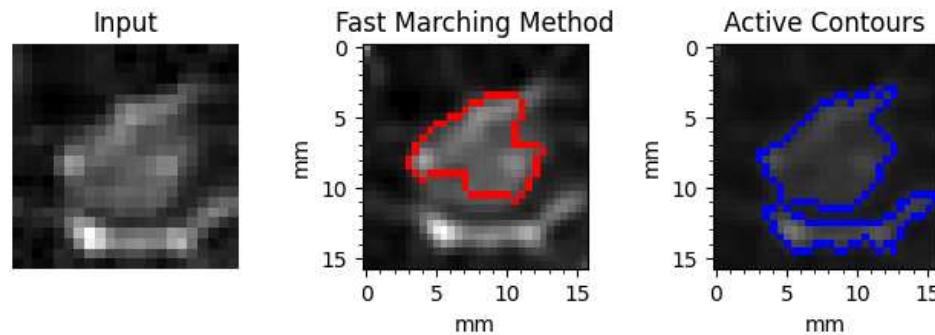
2: LIDC-IDRI-0179-15.58mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



3: LIDC-IDRI-0179-8.26mm

	AP	SP	ODQ
FMM	3	4	3
AC	5	5	5



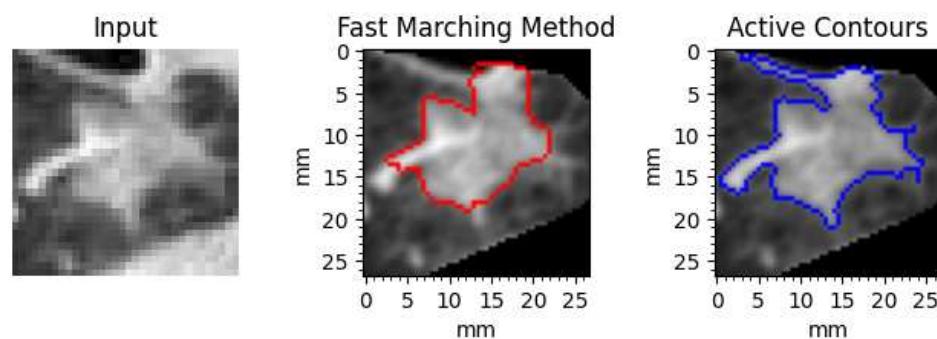
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

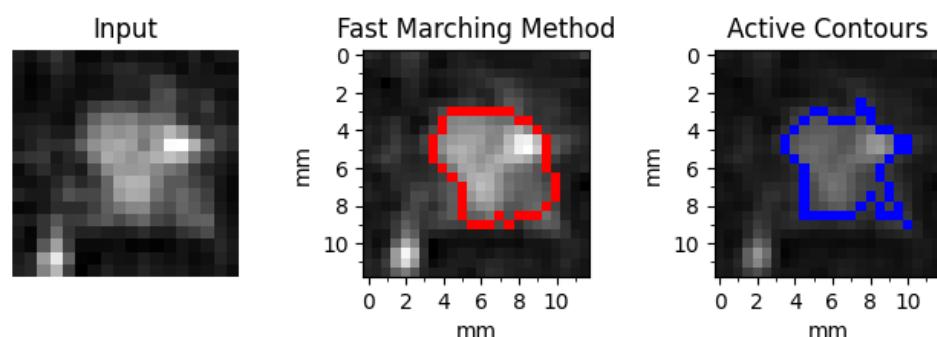
4: LIDC-IDRI-0220-13.53mm

	AP	SP	ODQ
FMM	4	5	5
AC	5	5	5



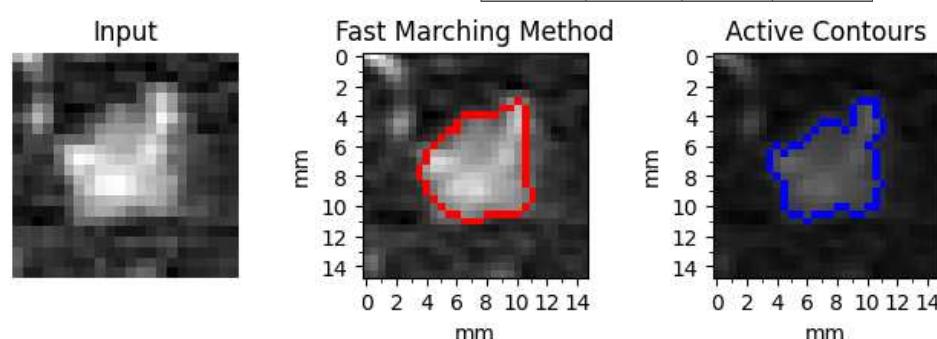
5: LIDC-IDRI-0298-6.31mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



6: LIDC-IDRI-0298-7.7mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



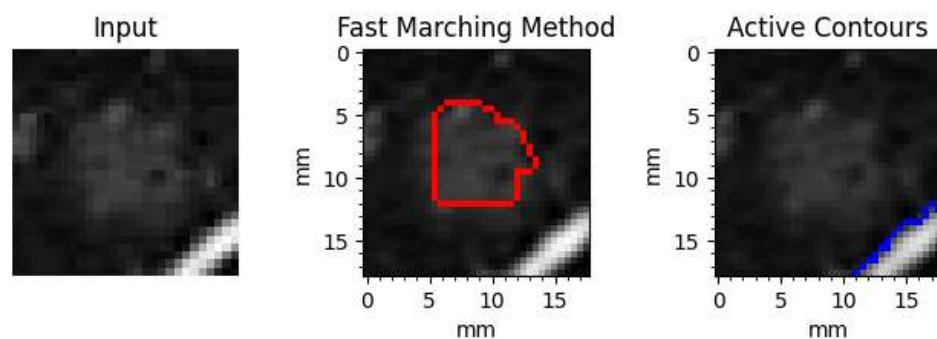
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

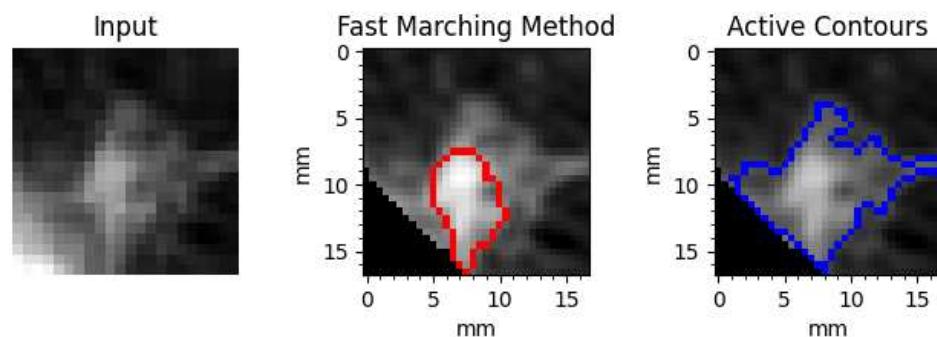
7: LIDC-IDRI-0298-9.41mm

	AP	SP	ODQ
FMM	3	3	3
AC	1	1	1



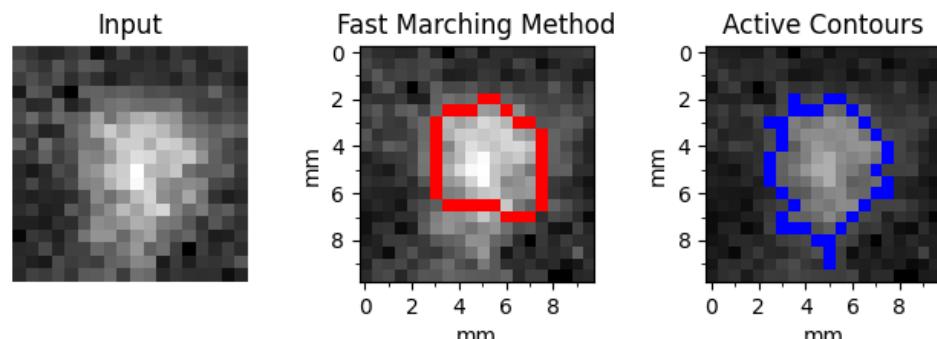
8: LIDC-IDRI-0311-8.81mm

	AP	SP	ODQ
FMM	2	2	3
AC	5	5	5



9: LIDC-IDRI-0394-5.1mm

	AP	SP	ODQ
FMM	3	3	3
AC	5	5	5



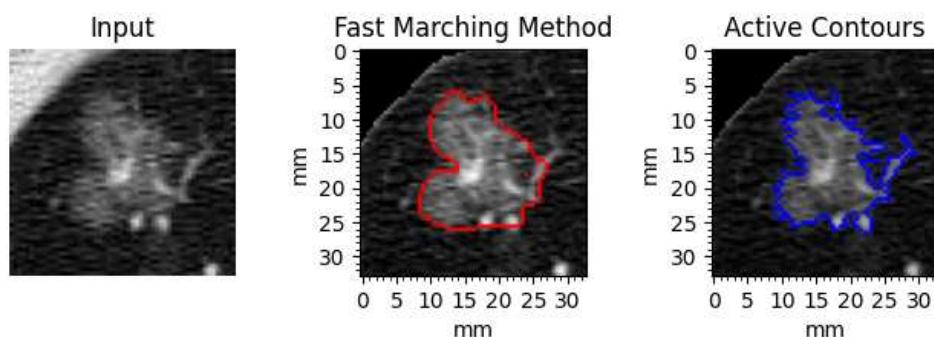
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

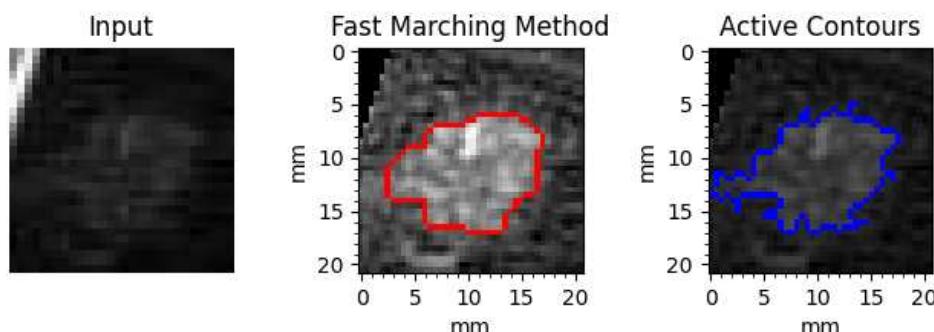
10: LIDC-IDRI-0464-16.72mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



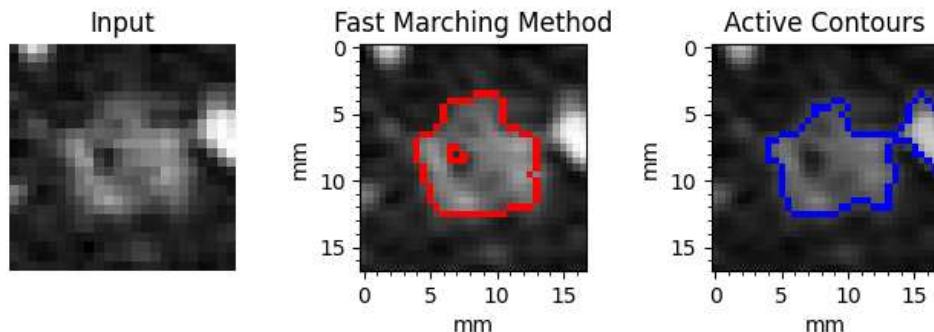
11: LIDC-IDRI-0497-10.61mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



12: LIDC-IDRI-0507-8.69mm

	AP	SP	ODQ
FMM	5	5	5
AC	4	5	5



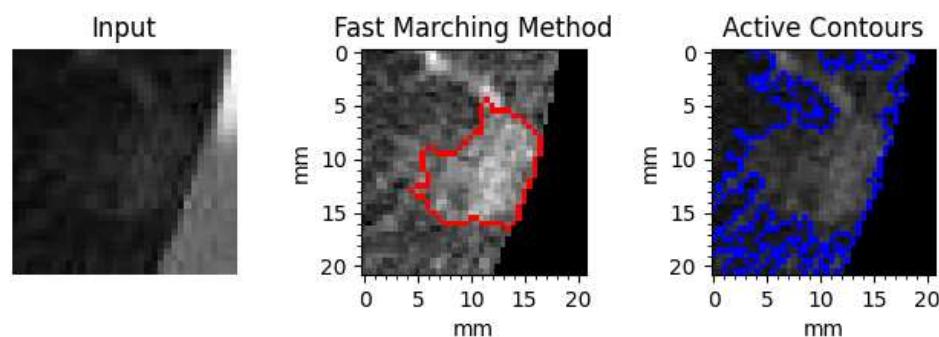
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

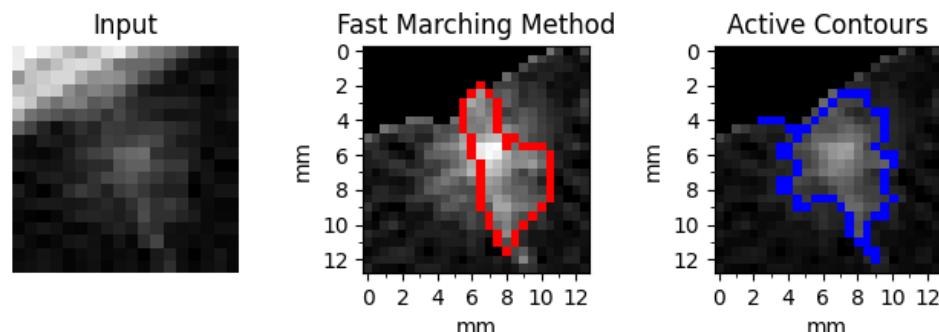
13: LIDC-IDRI-0660-10.52mm

	AP	SP	ODQ
FMM	4	5	5
AC	5	5	5



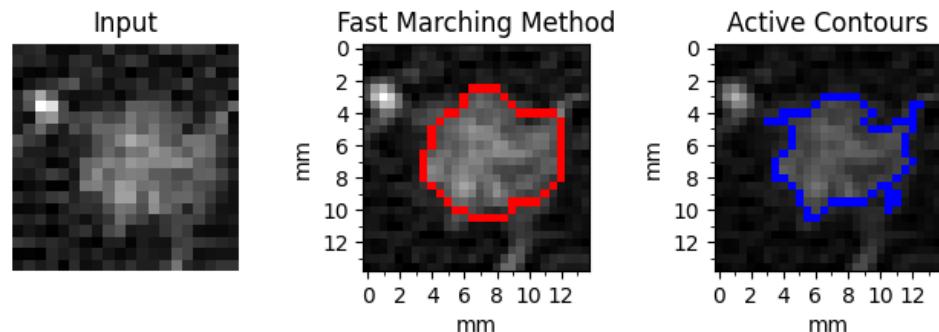
14: LIDC-IDRI-0686-6.73mm

	AP	SP	ODQ
FMM	3	4	4
AC	5	5	5



15: LIDC-IDRI-0686-7.21mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



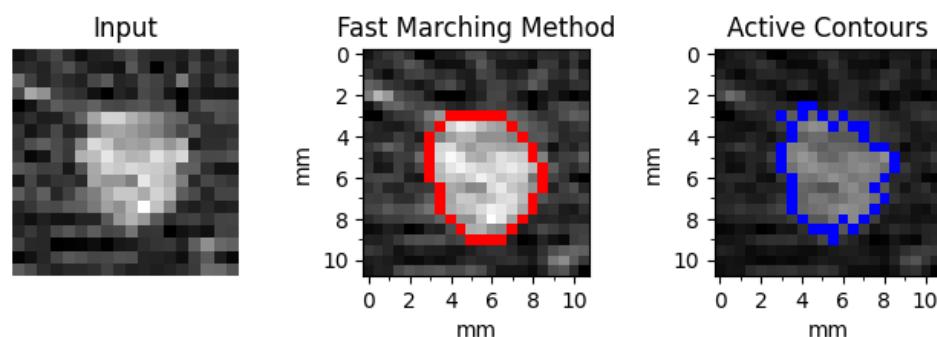
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

16: LIDC-IDRI-0759-5.84mm

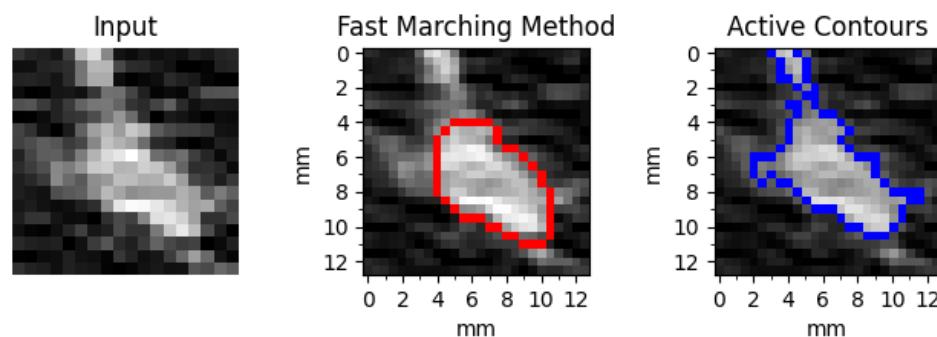
	AP	SP	ODQ
FMM	5	5	5
AC	5	5	5



---

17: LIDC-IDRI-0816-6.54mm

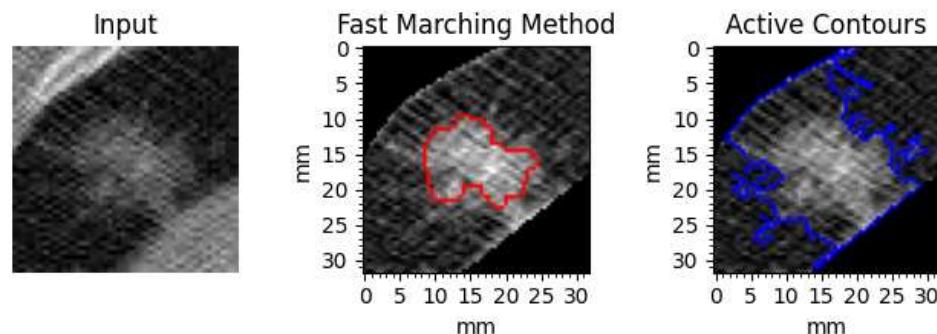
	AP	SP	ODQ
FMM	4	5	4
AC	5	5	5



---

18: LIDC-IDRI-0850-16.18mm

	AP	SP	ODQ
FMM	5	5	5
AC	5	3	3



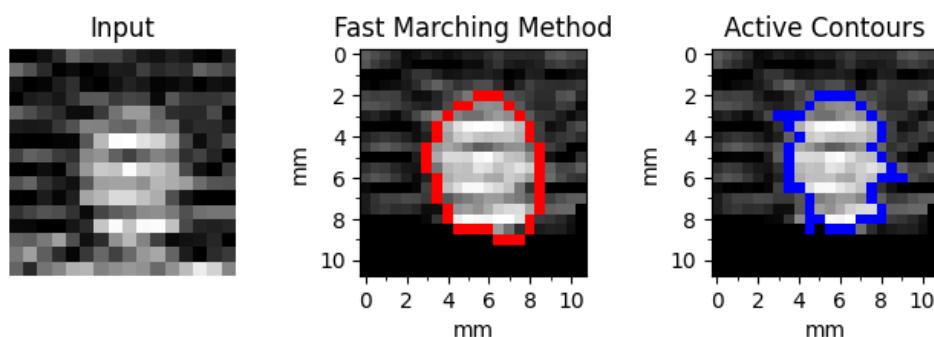
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

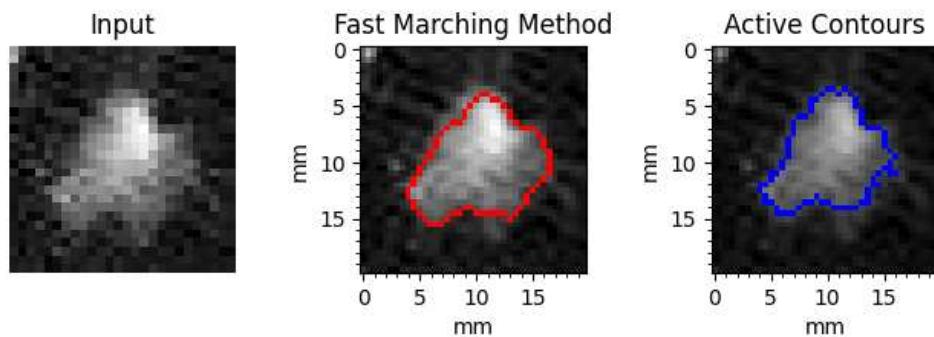
19: LIDC-IDRI-0894-5.98mm

	AP	SP	ODQ
FMM	5	5	4
AC	5	5	4



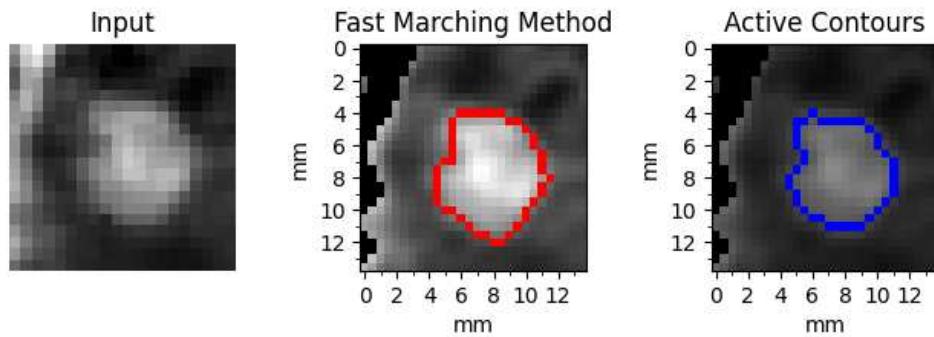
20: LIDC-IDRI-0938-10.38mm

	AP	SP	ODQ
FMM	4	5	5
AC	5	5	5



21: LIDC-IDRI-0966-7.12mm

	AP	SP	ODQ
FMM	4	5	4
AC	5	5	5



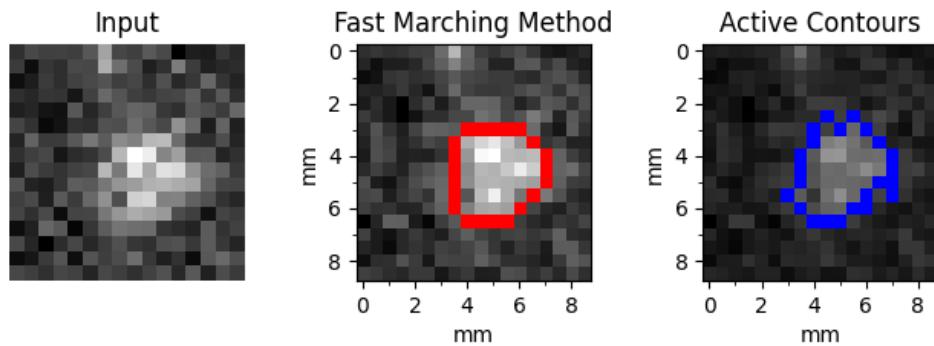
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

22: LIDC-IDRI-0984-4.89mm

	AP	SP	ODQ
FMM	4	4	4
AC	5	5	5



---

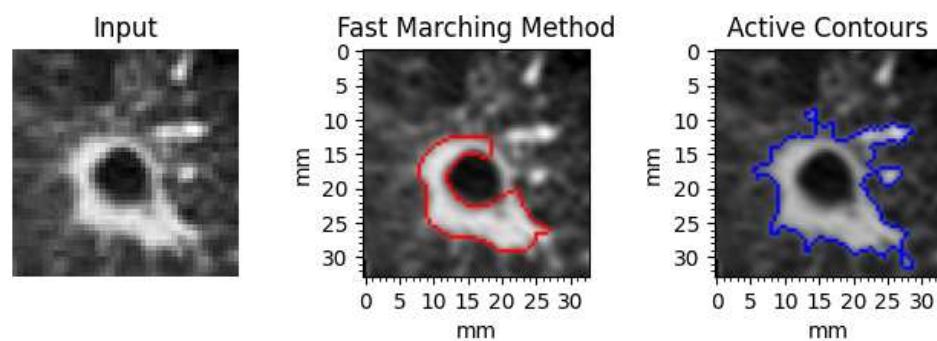
APPENDIX A. SEGMENTATION RESULTS - LIDC

---

#### A.4 Cavitary

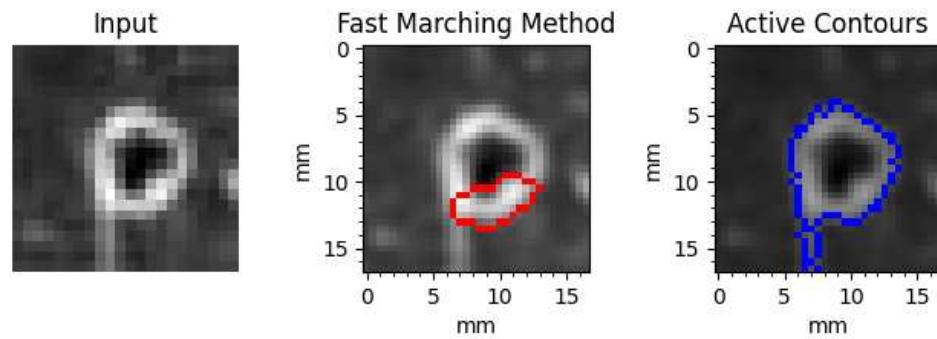
1: LIDC-IDRI-0052-16.98mm

	AP	SP	ODQ
FMM	4	4	4
AC	4	4	5



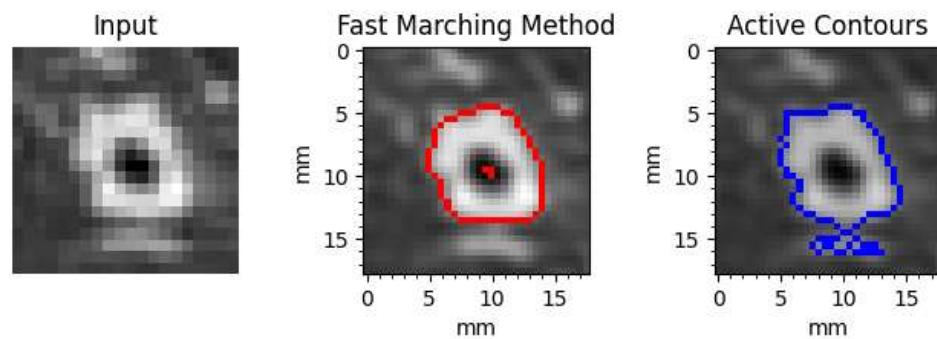
2: LIDC-IDRI-0063-8.77mm

	AP	SP	ODQ
FMM	1	1	1
AC	4	4	4



3: LIDC-IDRI-0091-9.26mm

	AP	SP	ODQ
FMM	5	5	5
AC	4	4	5



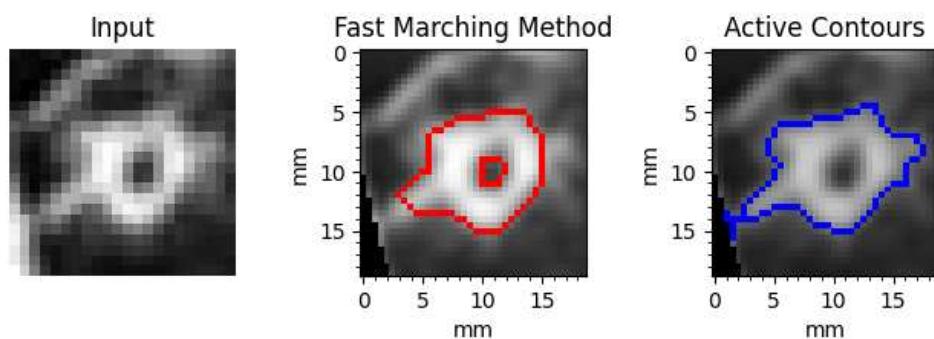
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

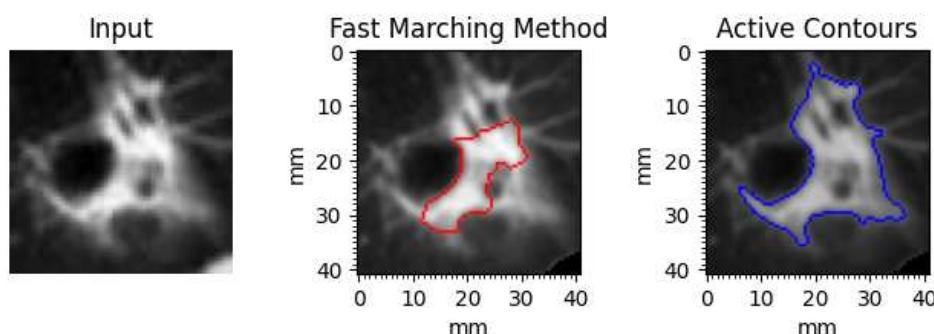
4: LIDC-IDRI-0114-9.75mm

	AP	SP	ODQ
FMM	5	5	5
AC	4	4	5



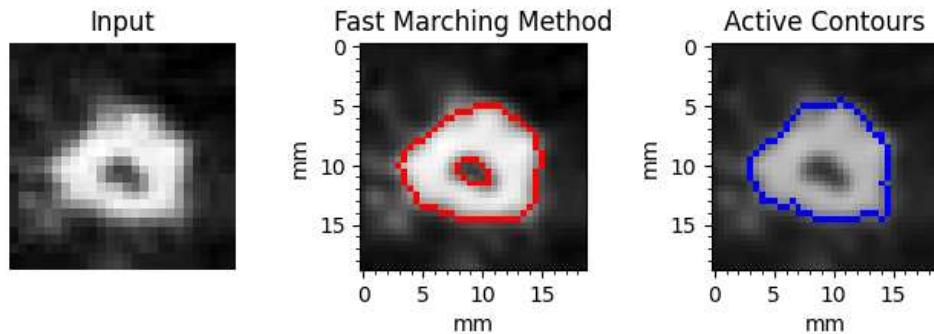
5: LIDC-IDRI-0194-20.52mm

	AP	SP	ODQ
FMM	4	4	5
AC	4	4	5



6: LIDC-IDRI-0206-9.82mm

	AP	SP	ODQ
FMM	5	5	5
AC	4	4	4



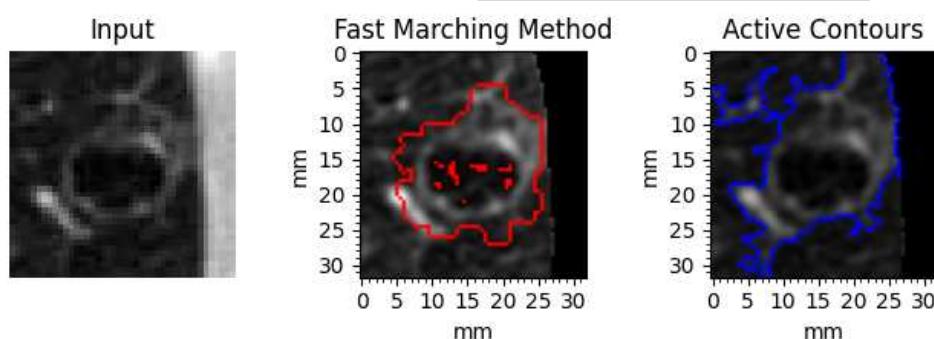
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

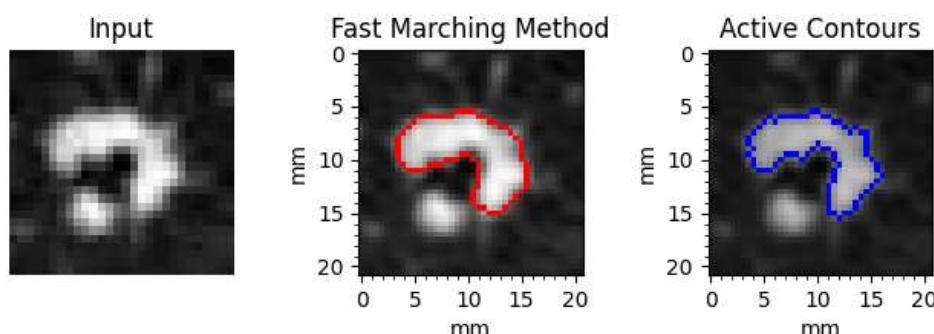
7: LIDC-IDRI-0681-16.1mm

	AP	SP	ODQ
FMM	5	5	5
AC	4	4	5



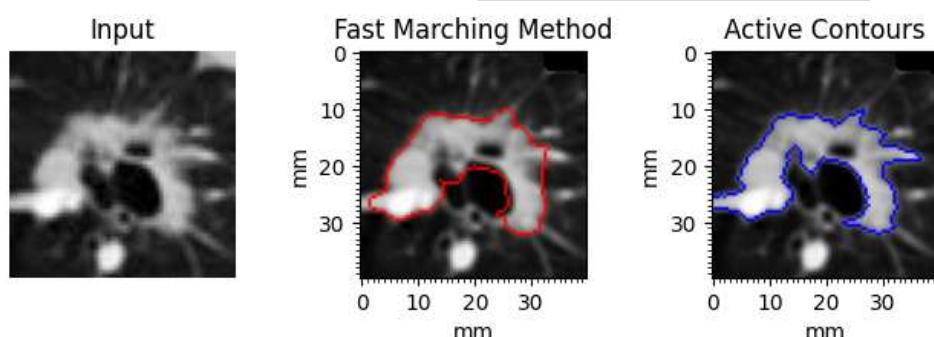
8: LIDC-IDRI-0844-10.72mm

	AP	SP	ODQ
FMM	4	4	3
AC	4	4	4



9: LIDC-IDRI-0865-20.12mm

	AP	SP	ODQ
FMM	4	4	4
AC	4	4	5



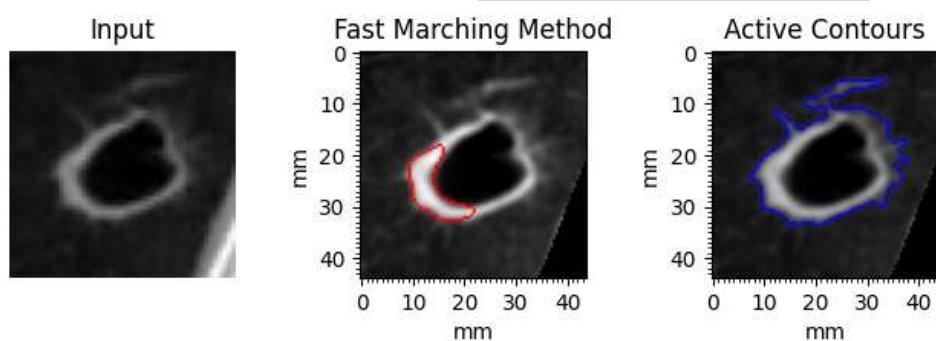
---

APPENDIX A. SEGMENTATION RESULTS - LIDC

---

10: LIDC-IDRI-1011-22.14mm

	AP	SP	ODQ
FMM	2	2	1
AC	5	5	5



## Appendix B

### Segmentation results - Phantom

Following are all the results obtained from using the phantom dataset. They are organized by category, and for each nodule there are two separate segmentation instances, one for FMM and one for AC.

The nodules are indexed in the same way as in the Evaluation section.

The objective evaluation results are displayed in a table containing, for both segmentations : intersection over union (IoU), long-axis diameter relative error (Long-axis err.) and short-axis diameter relative error (Short-axis err.). For each nodule, the segmentation boundaries obtained by both methods are displayed in a figure with three images, where the first image displays the ground truth boundary. Each image also has two dashed lines showing where long-axis and short-axis diameters were measured.

---

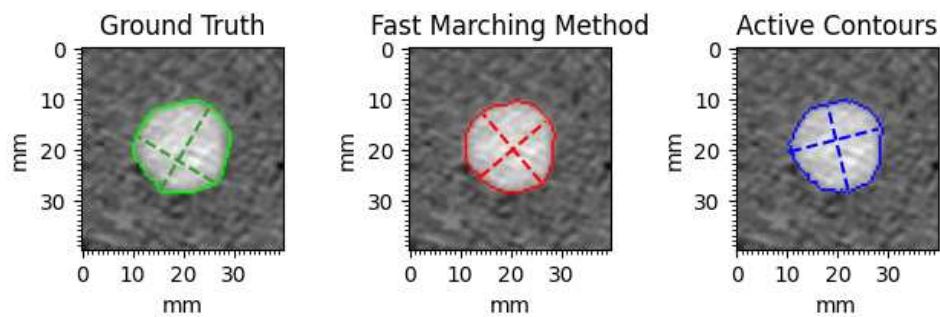
APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

### B.1 Round

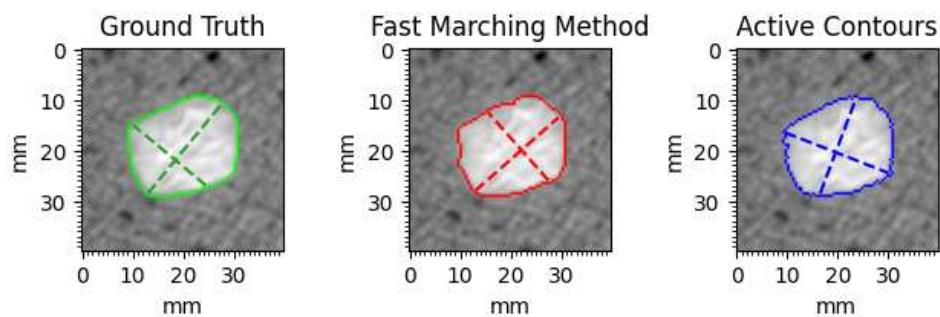
**1**

	IoU	Long-axis err.	Short-axis err.
FMM	0.930	4.73%	2.69%
AC	0.928	2.90%	2.97%



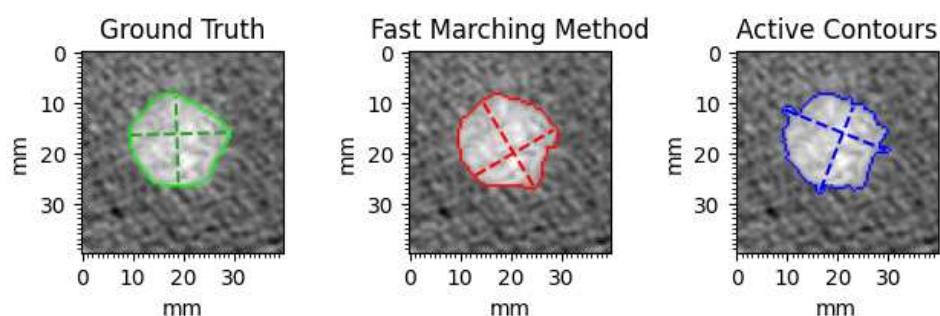
**2**

	IoU	Long-axis err.	Short-axis err.
FMM	0.945	3.34%	4.04%
AC	0.951	3.17%	5.10%



**3**

	IoU	Long-axis err.	Short-axis err.
FMM	0.929	1.47%	5.41%
AC	0.907	13.3%	4.35%



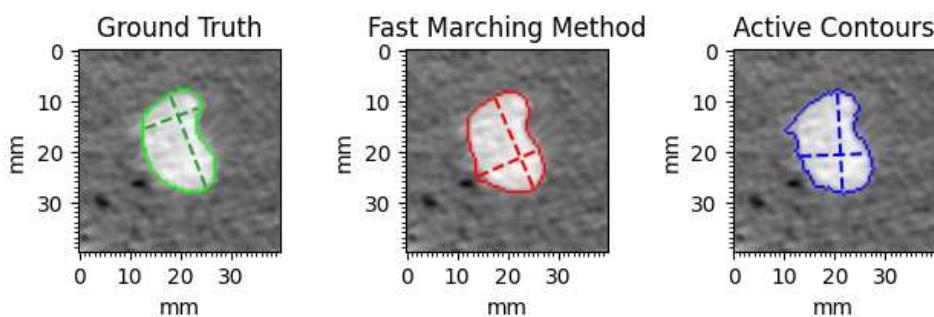
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

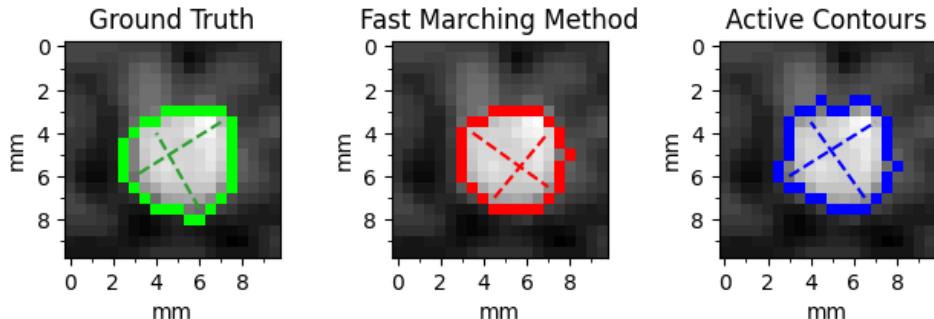
**4**

	IoU	Long-axis err.	Short-axis err.
FMM	0.902	1.40%	11.6%
AC	0.887	1.14%	15.7%



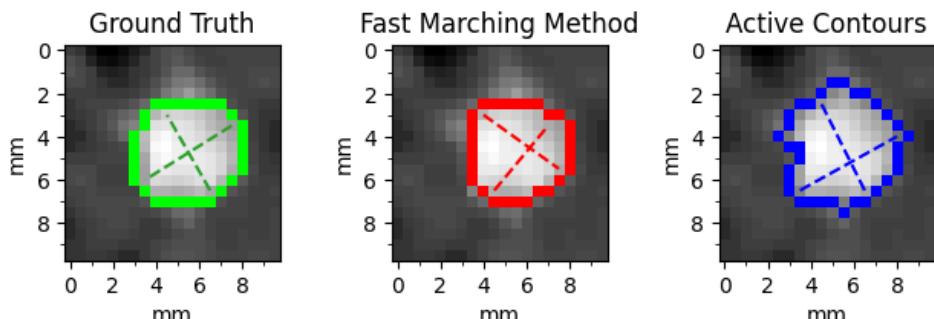
**5**

	IoU	Long-axis err.	Short-axis err.
FMM	0.851	8.90%	3.23%
AC	0.819	0%	6.70%



**6**

	IoU	Long-axis err.	Short-axis err.
FMM	0.877	8.90%	3.23%
AC	0.808	9.11%	10.9%



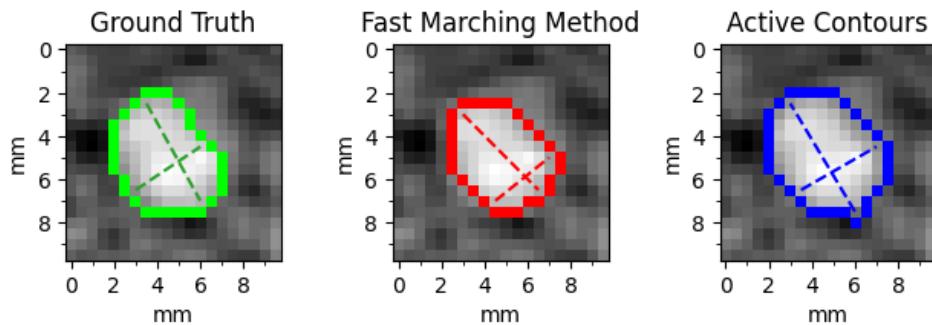
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

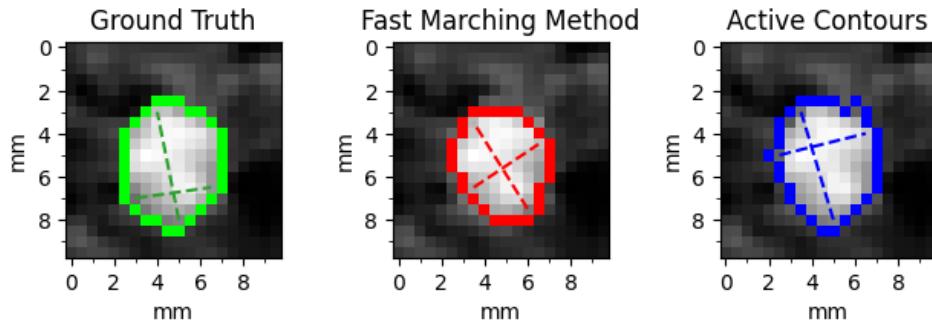
**7**

	IoU	Long-axis err.	Short-axis err.
FMM	0.743	3.88%	11.1%
AC	0.807	13.2%	11.9%



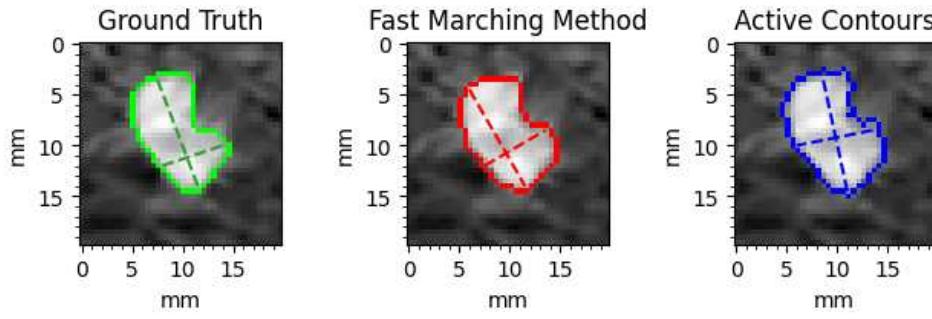
**8**

	IoU	Long-axis err.	Short-axis err.
FMM	0.778	7.45%	1.70%
AC	0.907	2.35%	16.4%



**9**

	IoU	Long-axis err.	Short-axis err.
FMM	0.885	2.31%	9.81%
AC	0.863	4.71%	21.0%



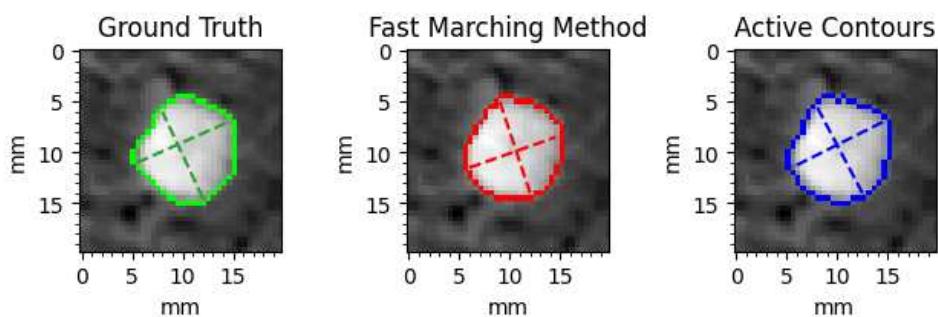
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

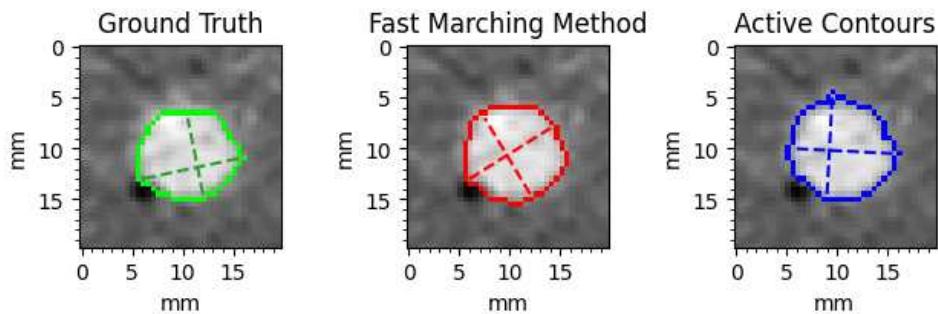
10

	IoU	Long-axis err.	Short-axis err.
FMM	0.913	3.65%	4.04%
AC	0.926	2.13%	2.34%



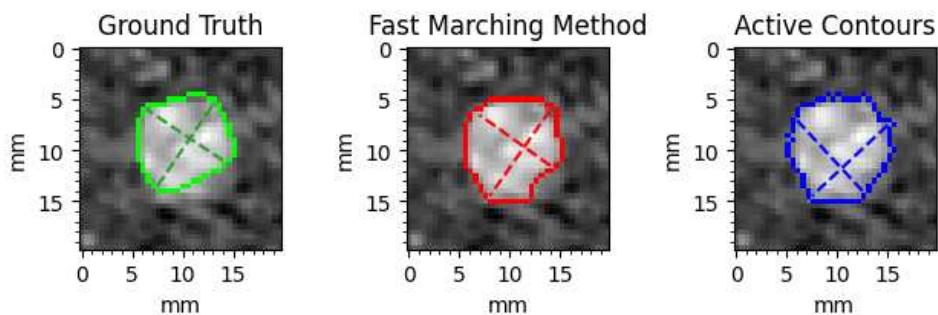
11

	IoU	Long-axis err.	Short-axis err.
FMM	0.908	2.88%	14.2%
AC	0.830	3.09%	24.4%



12

	IoU	Long-axis err.	Short-axis err.
FMM	0.853	7.11%	4.66%
AC	0.804	5.66%	5.88%



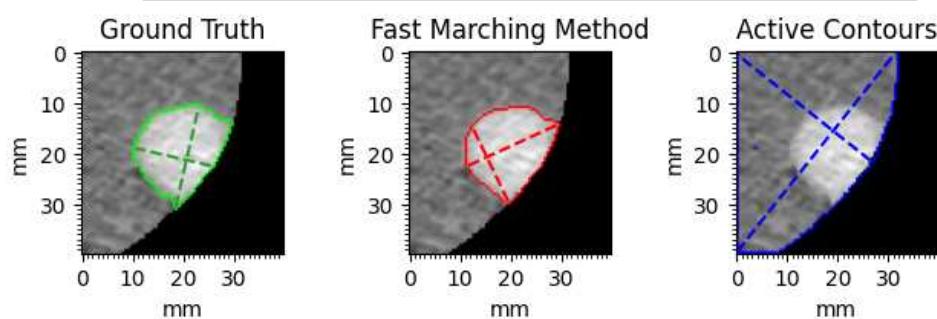
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

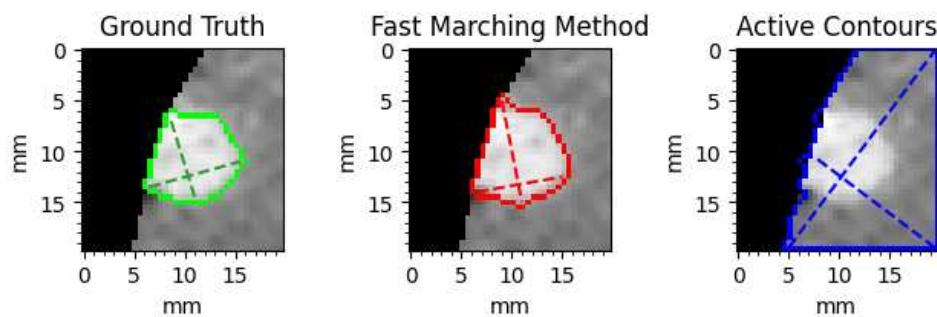
**13**

	IoU	Long-axis err.	Short-axis err.
FMM	0.942	3.85%	4.54%
AC	0.235	152%	120%



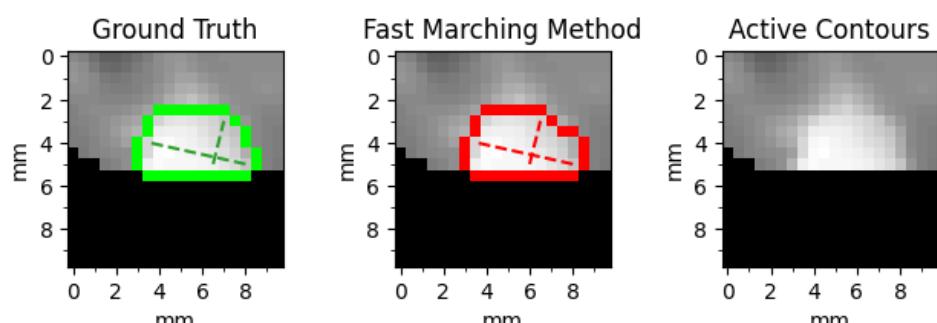
**14**

	IoU	Long-axis err.	Short-axis err.
FMM	0.899	9.21%	2.98%
AC	0.233	160%	79.0%



**15**

	IoU	Long-axis err.	Short-axis err.
FMM	0.933	0%	0%
AC	0	100%	100%



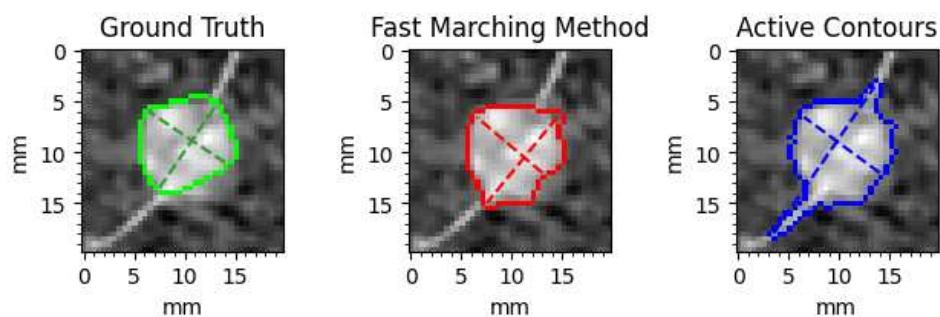
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

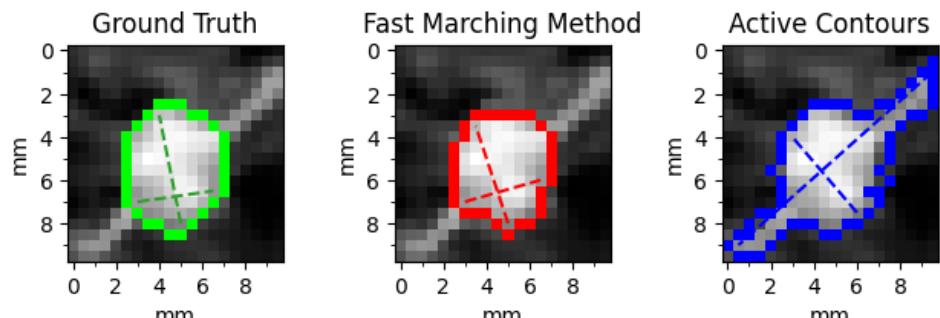
**16**

	IoU	Long-axis err.	Short-axis err.
FMM	0.797	13.4%	5.54%
AC	0.734	81.4%	3.10%



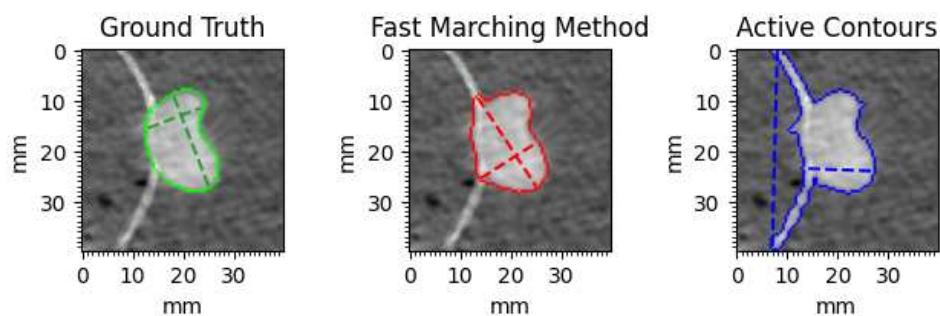
**17**

	IoU	Long-axis err.	Short-axis err.
FMM	0.875	7.06%	2.83%
AC	0.673	136%	30.2%



**18**

	IoU	Long-axis err.	Short-axis err.
FMM	0.866	8.44%	16.0%
AC	0.715	102%	20.2%



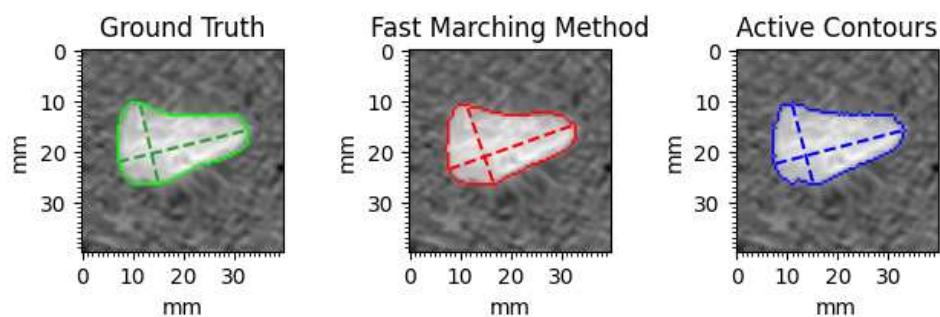
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

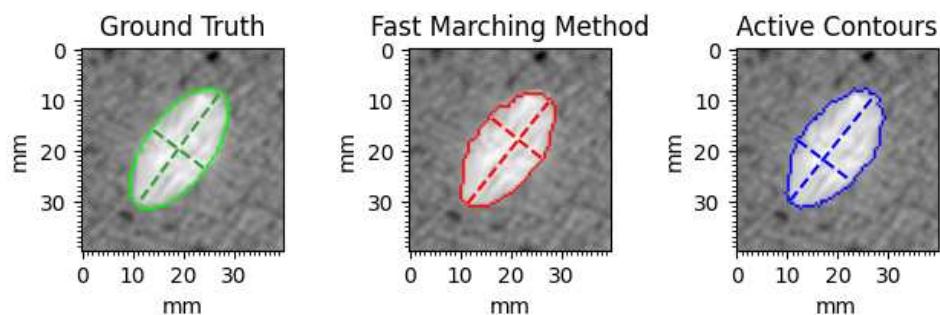
**19**

	IoU	Long-axis err.	Short-axis err.
FMM	0.936	0.952%	0.390%
AC	0.948	2.42%	0.779%



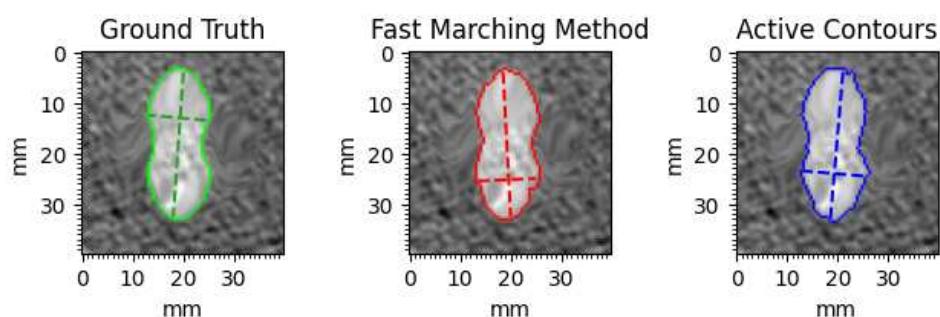
**20**

	IoU	Long-axis err.	Short-axis err.
FMM	0.916	4.44%	2.40%
AC	0.925	1.79%	5.60%



**21**

	IoU	Long-axis err.	Short-axis err.
FMM	0.930	0.105%	0.260%
AC	0.943	1.89%	4.33%



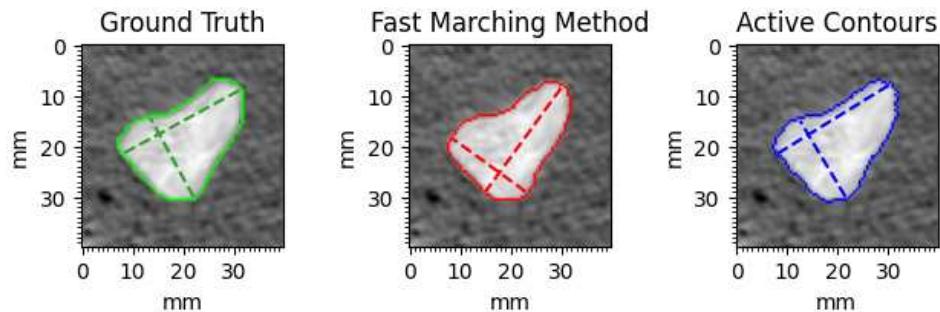
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

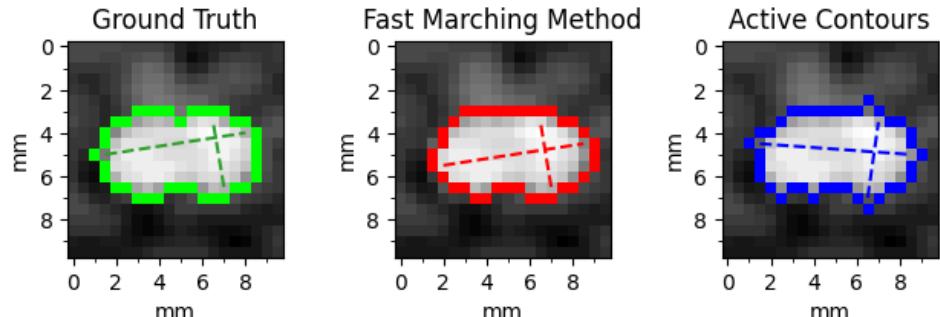
**22**

	IoU	Long-axis err.	Short-axis err.
FMM	0.940	2.98%	1.24%
AC	0.952	1.41%	1.02%



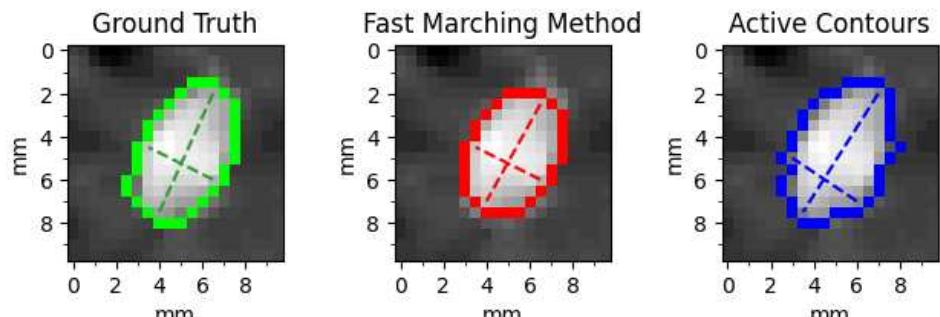
**23**

	IoU	Long-axis err.	Short-axis err.
FMM	0.893	0%	0%
AC	0.884	6.69%	16.4%



**24**

	IoU	Long-axis err.	Short-axis err.
FMM	0.865	14.7%	0%
AC	0.880	7.95%	7.14%



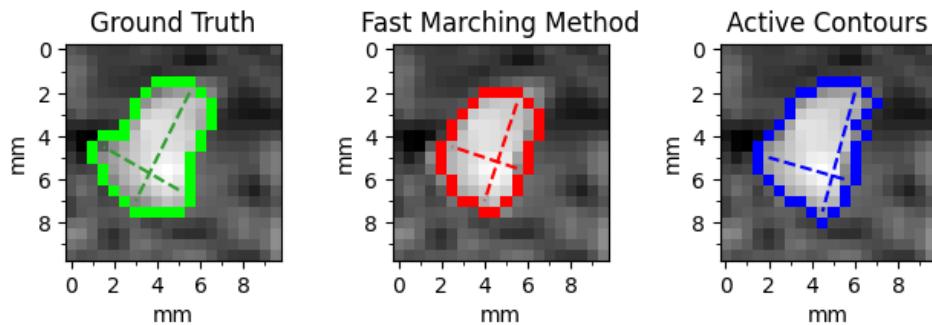
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

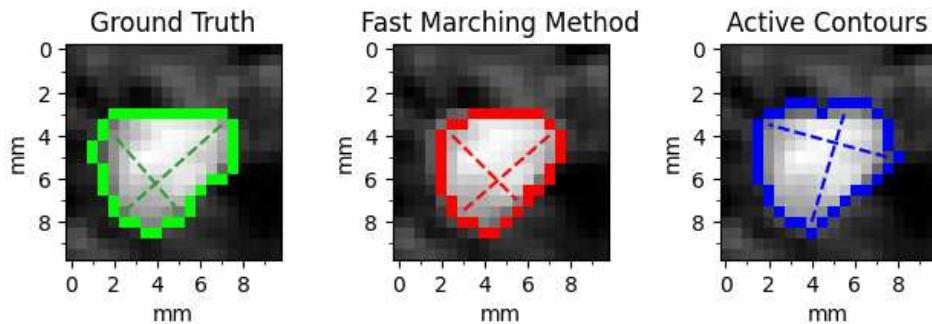
**25**

	IoU	Long-axis err.	Short-axis err.
FMM	0.730	15.2%	21.6%
AC	0.797	1.97%	9.68%



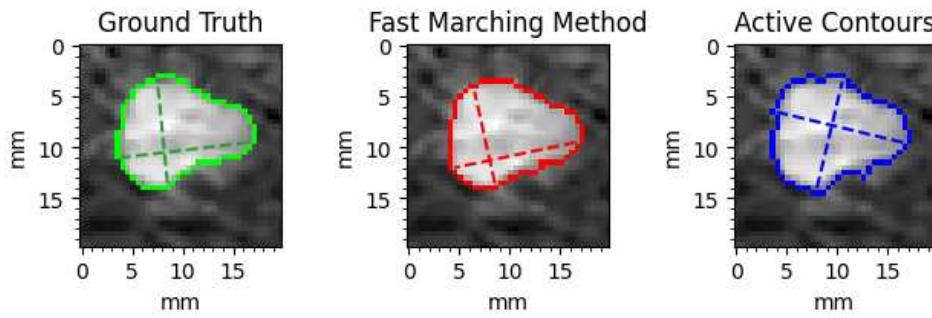
**26**

	IoU	Long-axis err.	Short-axis err.
FMM	0.791	11.6%	8.03%
AC	0.845	5.32%	13.2%



**27**

	IoU	Long-axis err.	Short-axis err.
FMM	0.911	2.62%	8.26%
AC	0.903	2.06%	7.46%



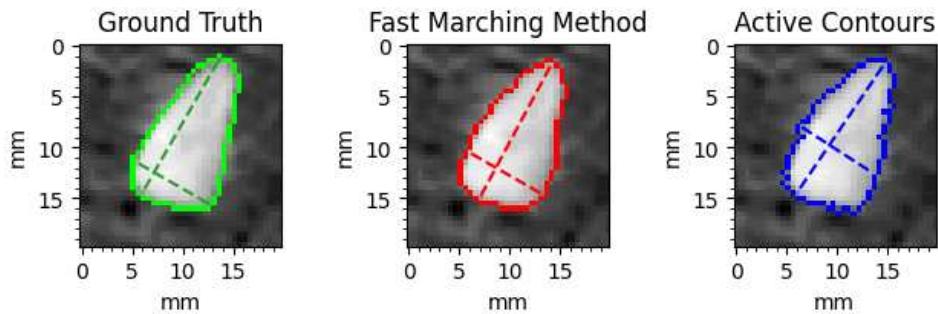
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

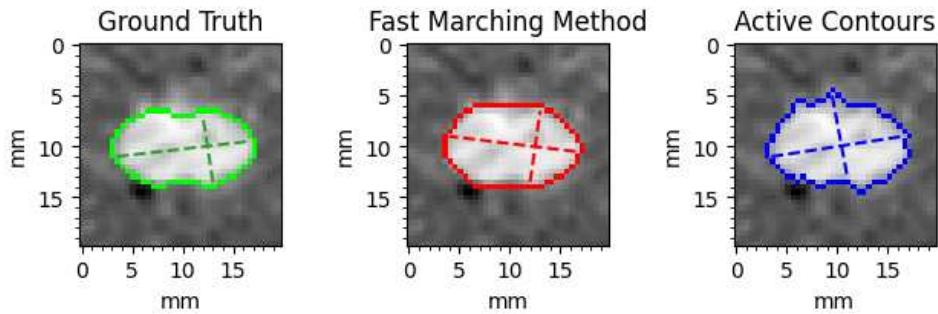
**28**

	IoU	Long-axis err.	Short-axis err.
FMM	0.921	1.67%	0%
AC	0.920	0.733%	3.23%



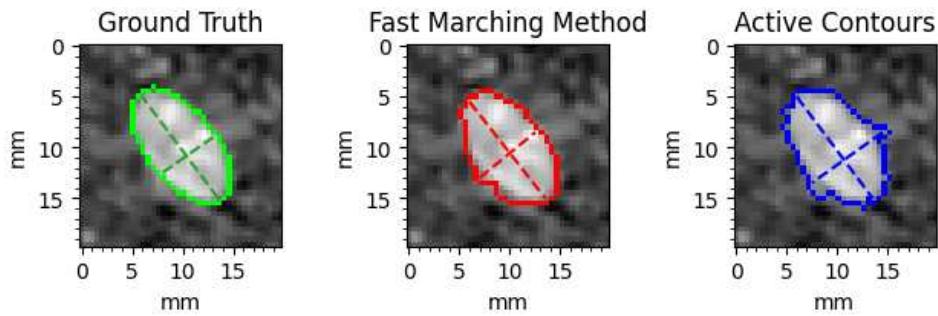
**29**

	IoU	Long-axis err.	Short-axis err.
FMM	0.868	3.82%	7.45%
AC	0.853	0.458%	23.7%



**30**

	IoU	Long-axis err.	Short-axis err.
FMM	0.894	3.20%	16.4%
AC	0.845	0%	41.0%



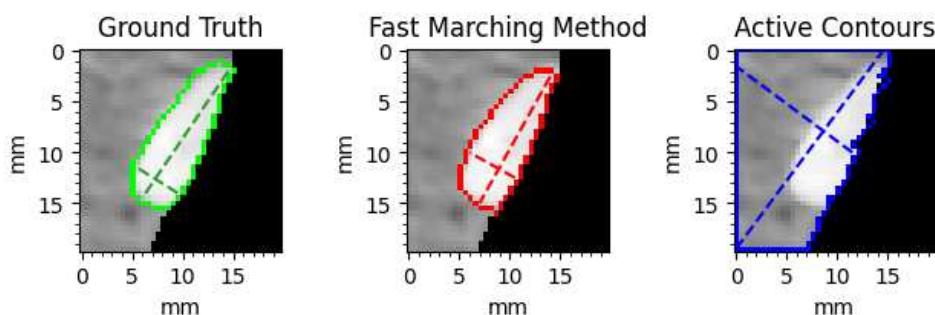
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

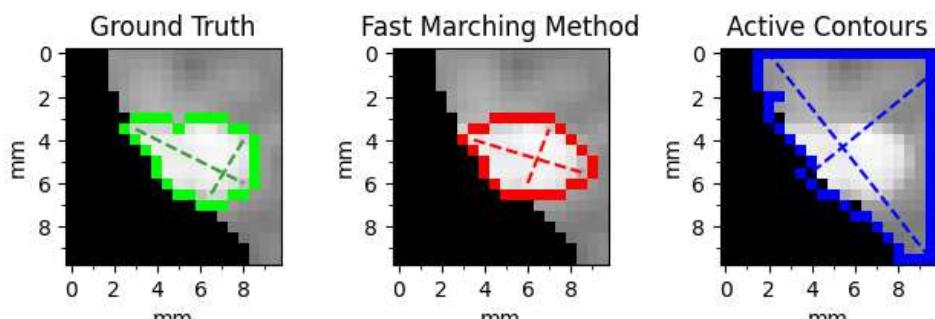
31

	IoU	Long-axis err.	Short-axis err.
FMM	0.919	5.22%	9.11%
AC	0.253	60.7%	203%



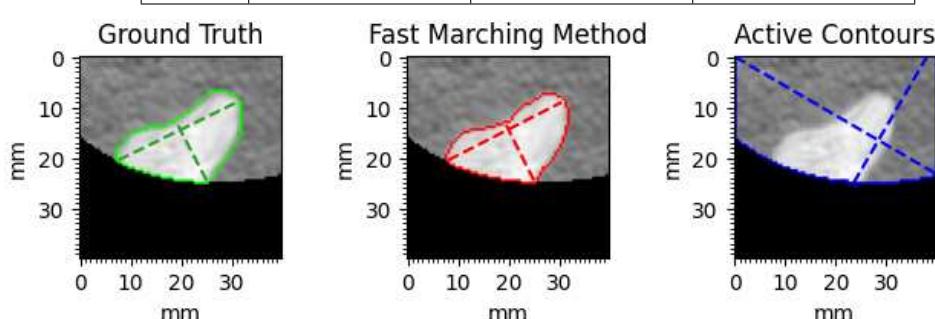
32

	IoU	Long-axis err.	Short-axis err.
FMM	0.804	6.62%	7.53%
AC	0.242	116%	143%



33

	IoU	Long-axis err.	Short-axis err.
FMM	0.933	3.40%	5.17%
AC	0.276	74.7%	123%



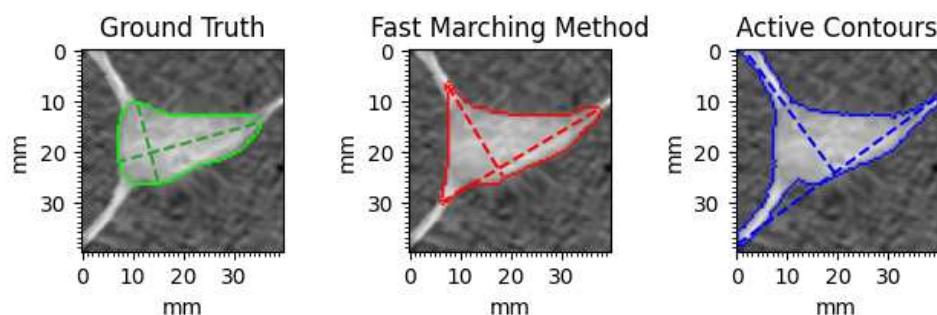
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

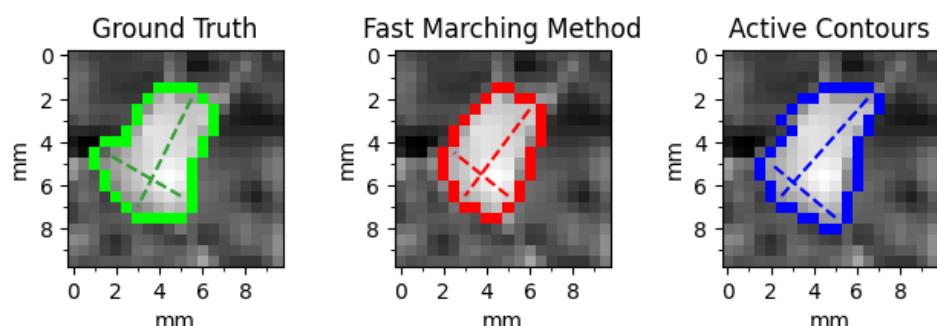
**34**

	IoU	Long-axis err.	Short-axis err.
FMM	0.856	26.4%	36.0%
AC	0.736	74.9%	95.9%



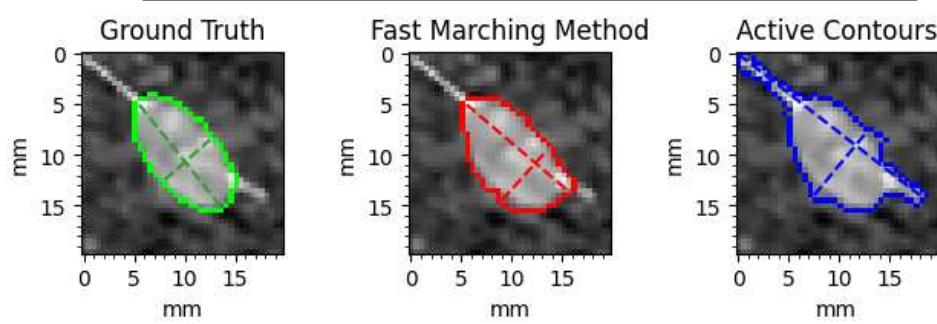
**35**

	IoU	Long-axis err.	Short-axis err.
FMM	0.770	10.6%	20.6%
AC	0.800	7.69%	3.23%



**36**

	IoU	Long-axis err.	Short-axis err.
FMM	0.882	2.42%	11.6%
AC	0.742	78.0%	29.7%



---

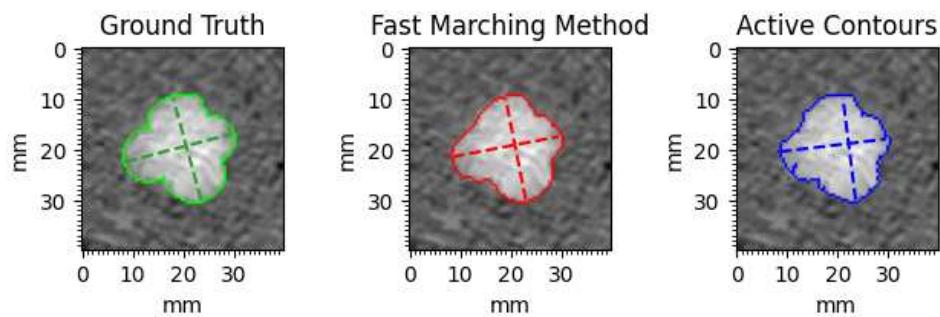
APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

## B.2 Irregular

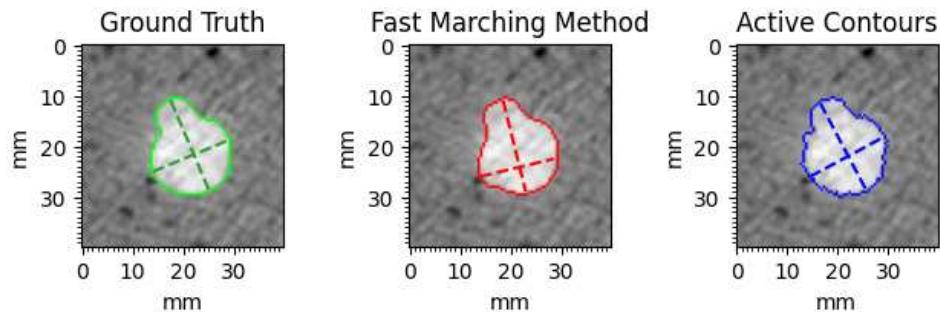
**1**

	IoU	Long-axis err.	Short-axis err.
FMM	0.946	1.52%	1.64%
AC	0.931	2.58%	2.80%



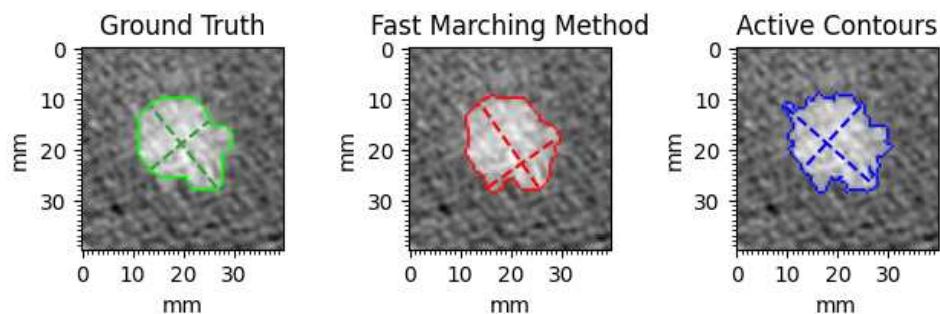
**2**

	IoU	Long-axis err.	Short-axis err.
FMM	0.950	2.57%	4.91%
AC	0.939	1.05%	1.21%



**3**

	IoU	Long-axis err.	Short-axis err.
FMM	0.908	2.47%	10.7%
AC	0.879	13.4%	13.8%



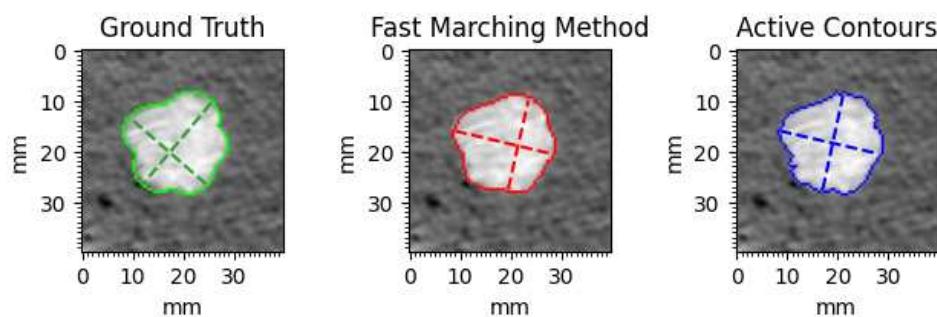
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

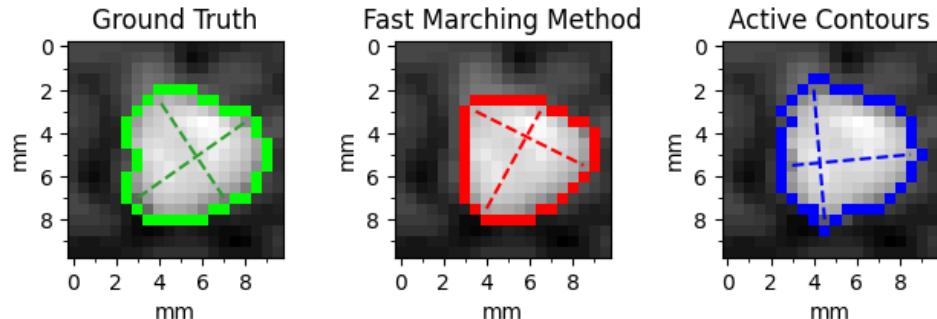
**4**

	IoU	Long-axis err.	Short-axis err.
FMM	0.923	6.69%	9.57%
AC	0.933	4.40%	4.68%



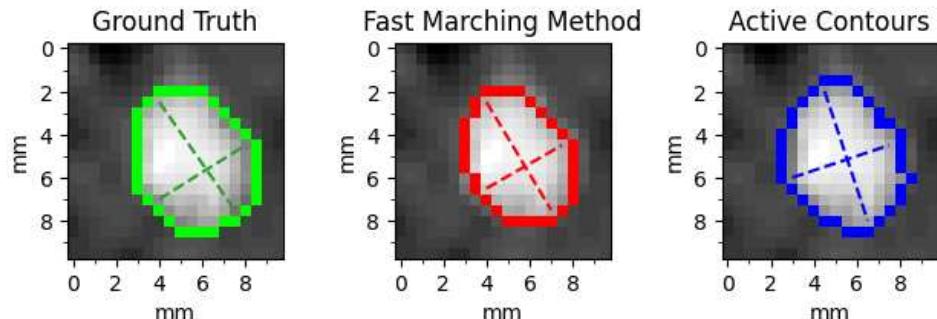
**5**

	IoU	Long-axis err.	Short-axis err.
FMM	0.867	8.36%	4.81%
AC	0.833	1.31%	2.03%



**6**

	IoU	Long-axis err.	Short-axis err.
FMM	0.798	4.43%	14.6%
AC	0.850	3.61%	0.424%



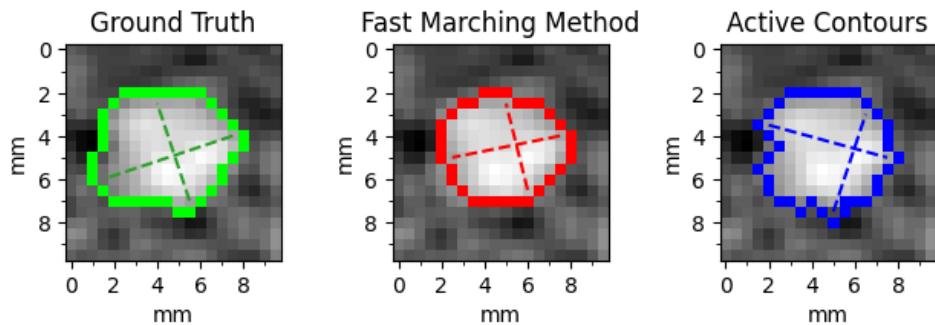
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

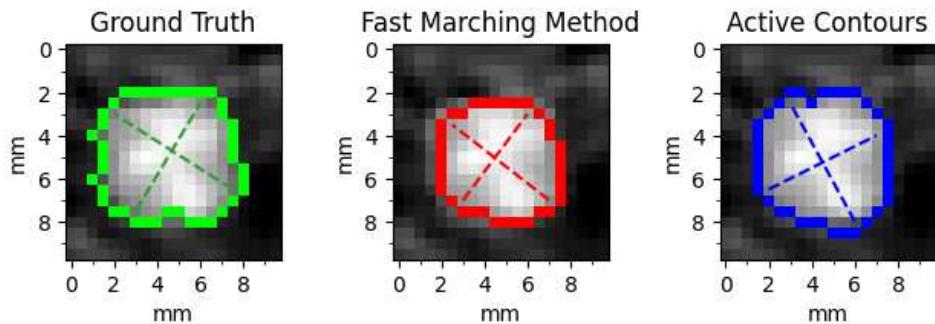
**7**

	IoU	Long-axis err.	Short-axis err.
FMM	0.750	19.3%	13.1%
AC	0.808	9.81%	0%



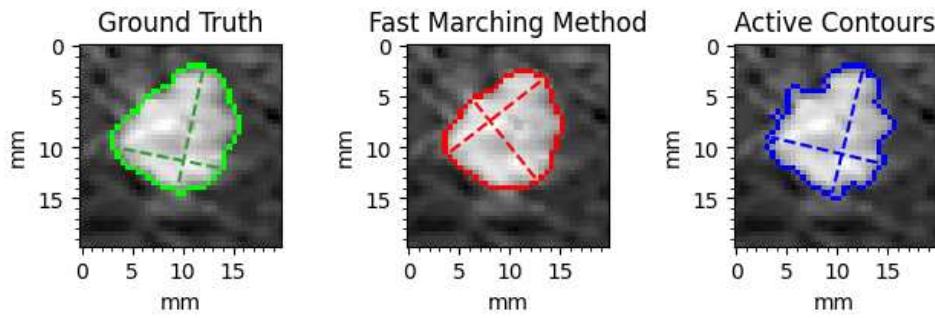
**8**

	IoU	Long-axis err.	Short-axis err.
FMM	0.708	12.6%	14.2%
AC	0.870	3.99%	4.12%



**9**

	IoU	Long-axis err.	Short-axis err.
FMM	0.906	3.14%	5.88%
AC	0.893	5.10%	5.88%



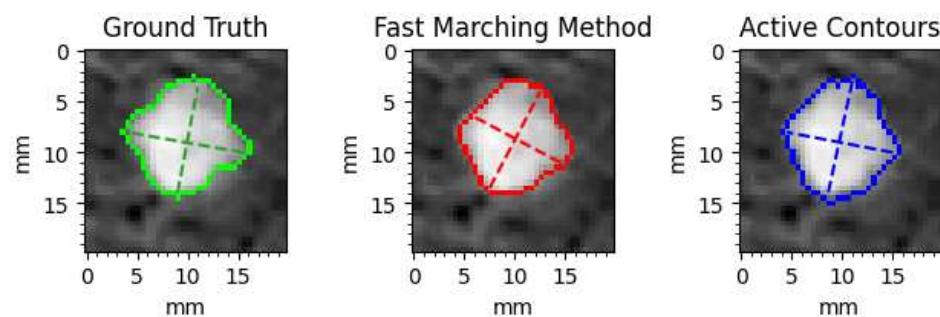
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

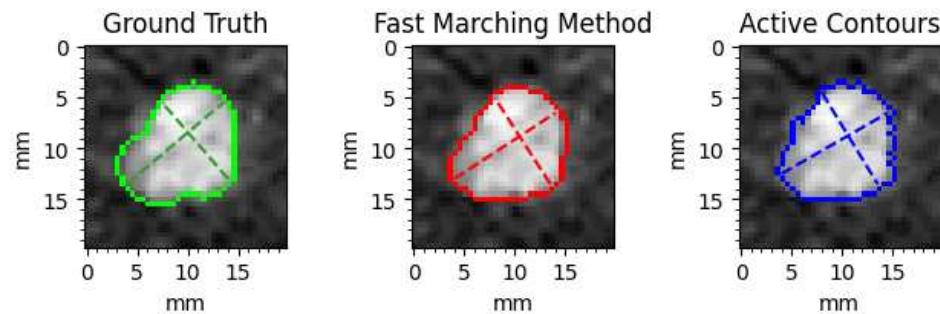
10

	IoU	Long-axis err.	Short-axis err.
FMM	0.905	7.97%	10.0%
AC	0.855	0.857%	0%



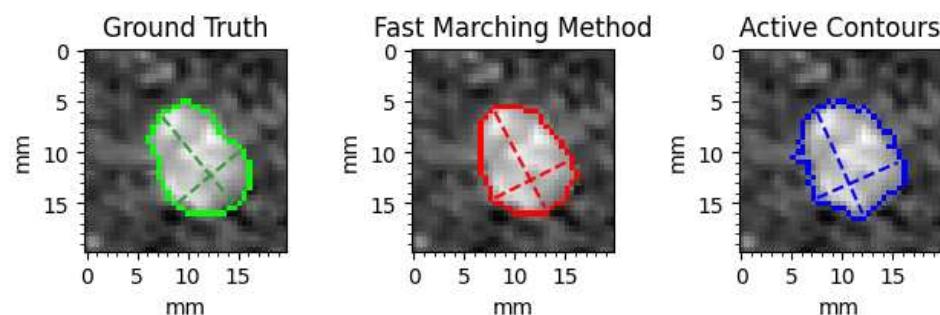
11

	IoU	Long-axis err.	Short-axis err.
FMM	0.897	3.94%	1.91%
AC	0.867	2.66%	6.24%



12

	IoU	Long-axis err.	Short-axis err.
FMM	0.900	2.45%	0.128%
AC	0.833	0.363%	11.8%



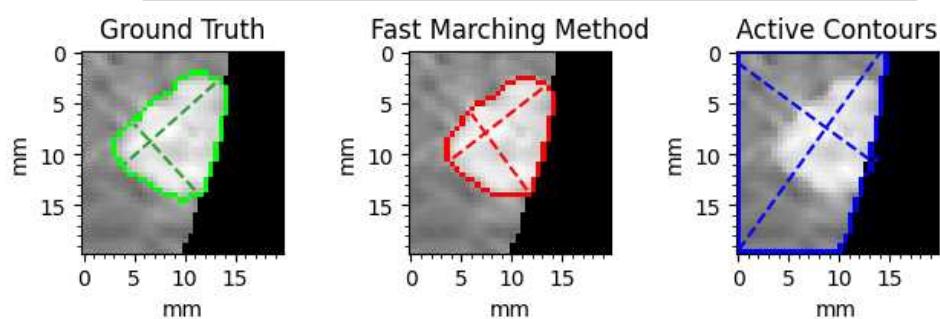
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

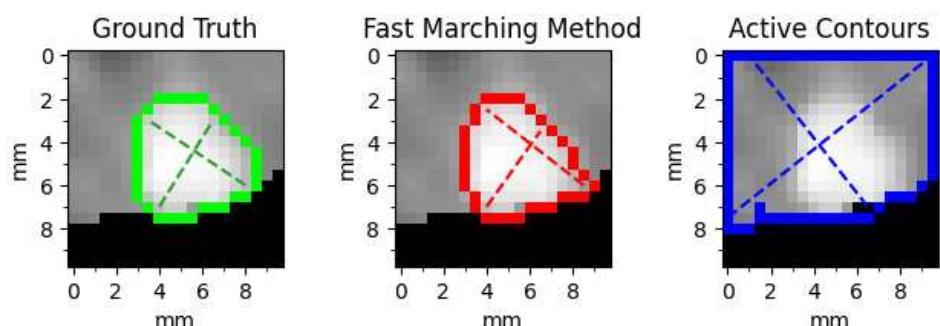
**13**

	IoU	Long-axis err.	Short-axis err.
FMM	0.931	5.32%	5.08%
AC	0.323	99.4%	81.9%



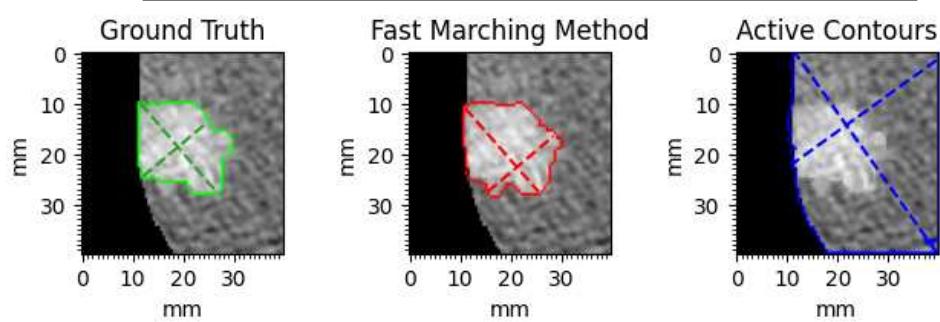
**14**

	IoU	Long-axis err.	Short-axis err.
FMM	0.890	5.36%	8.90%
AC	0.281	124%	88.6%



**15**

	IoU	Long-axis err.	Short-axis err.
FMM	0.892	4.68%	4.35%
AC	0.219	110%	115%



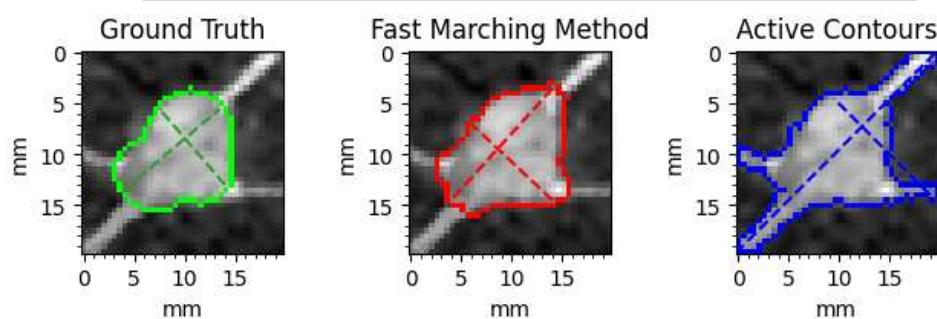
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

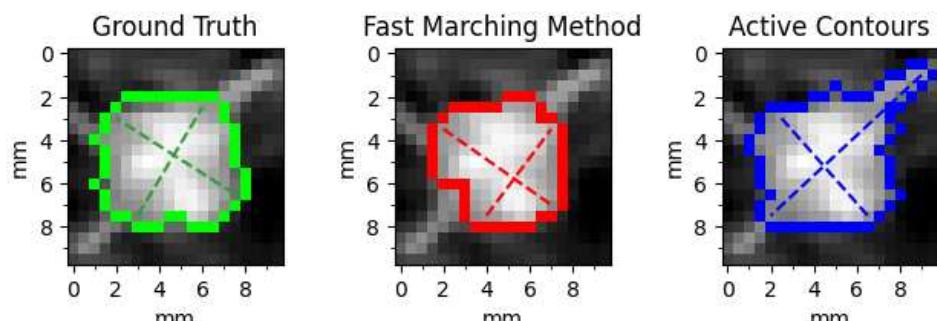
**16**

	IoU	Long-axis err.	Short-axis err.
FMM	0.829	19.7%	10.4%
AC	0.667	122%	35.3%



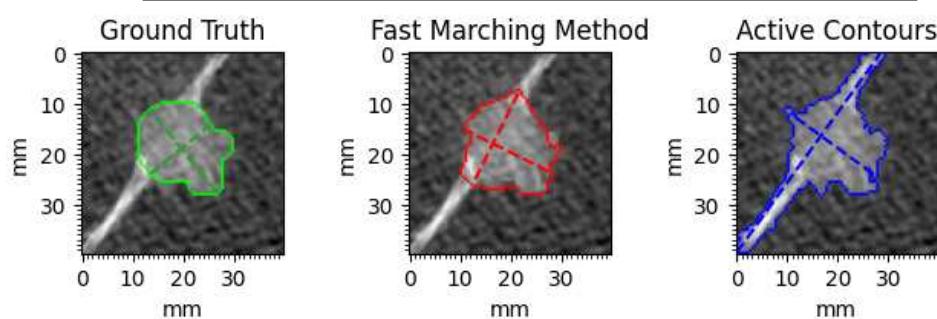
**17**

	IoU	Long-axis err.	Short-axis err.
FMM	0.783	6.44%	14.2%
AC	0.815	46.5%	3.26%



**18**

	IoU	Long-axis err.	Short-axis err.
FMM	0.824	2.42%	19.9%
AC	0.637	136%	36.7%



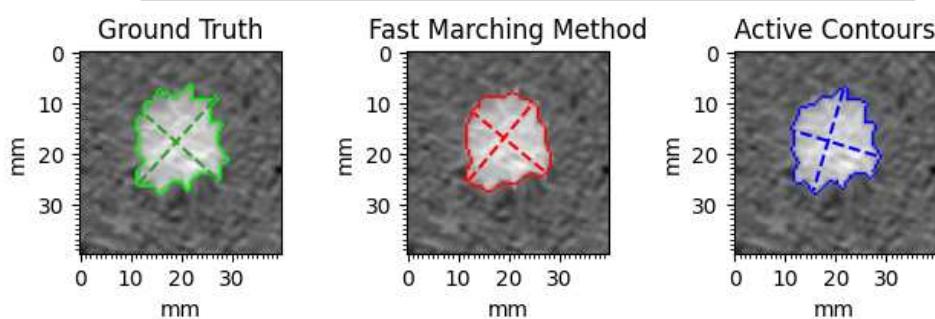
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

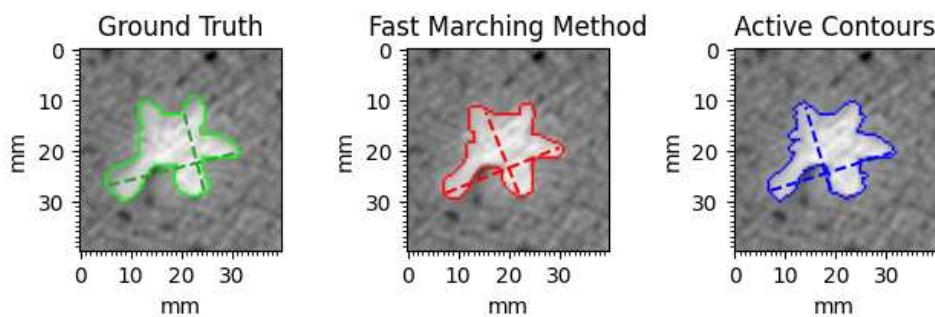
**19**

	IoU	Long-axis err.	Short-axis err.
FMM	0.906	12.4%	3.46%
AC	0.933	6.37%	8.15%



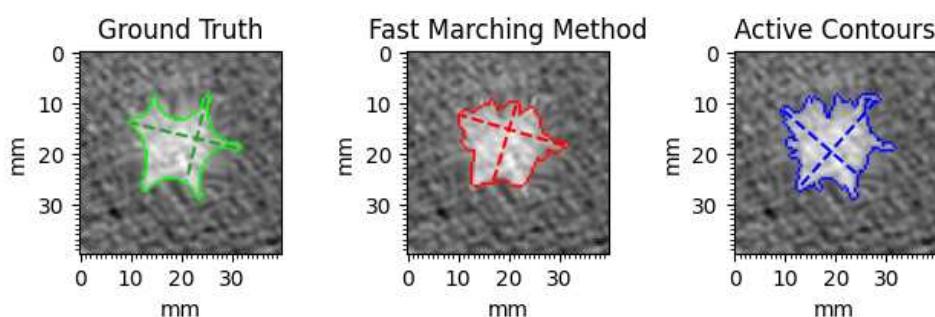
**20**

	IoU	Long-axis err.	Short-axis err.
FMM	0.805	7.94%	7.87%
AC	0.868	4.48%	7.68%



**21**

	IoU	Long-axis err.	Short-axis err.
FMM	0.776	1.06%	7.77%
AC	0.769	8.29%	12.7%



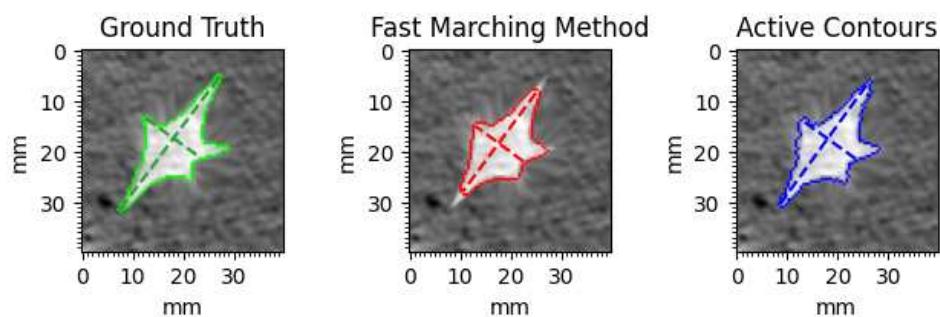
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

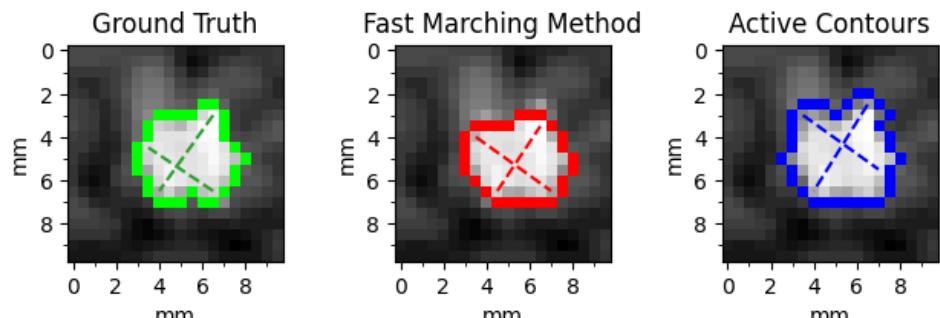
**22**

	IoU	Long-axis err.	Short-axis err.
FMM	0.846	25.2%	8.53%
AC	0.873	8.62%	0%



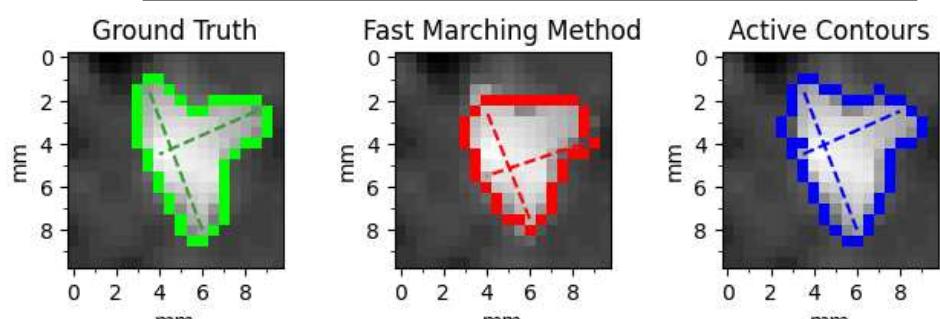
**23**

	IoU	Long-axis err.	Short-axis err.
FMM	0.804	0%	0%
AC	0.785	9.77%	19.4%



**24**

	IoU	Long-axis err.	Short-axis err.
FMM	0.764	22.6%	3.66%
AC	0.830	0%	0%



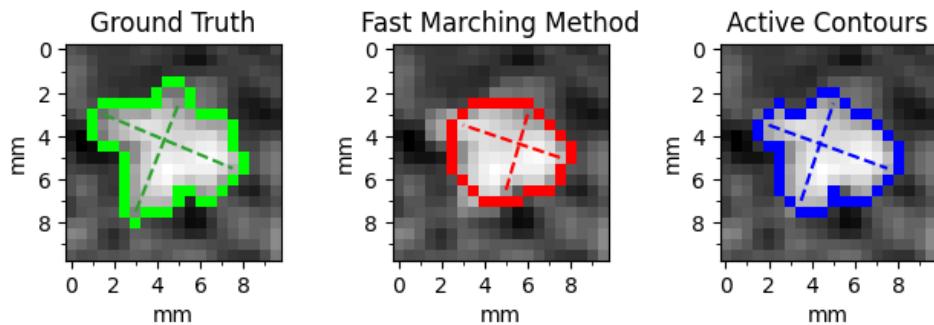
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

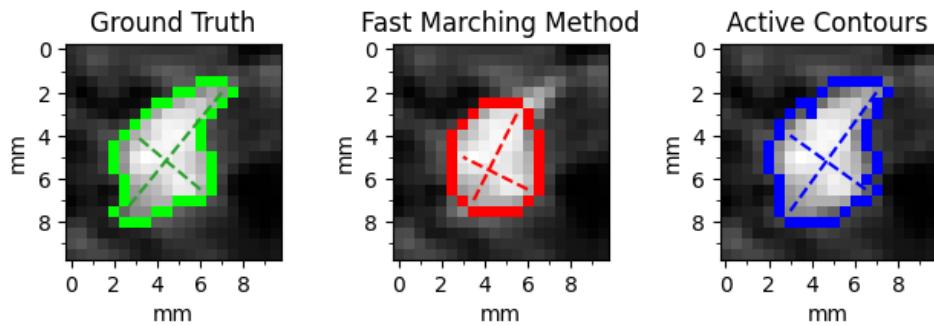
**25**

	IoU	Long-axis err.	Short-axis err.
FMM	0.659	27.1%	32.3%
AC	0.843	10.0%	11.9%



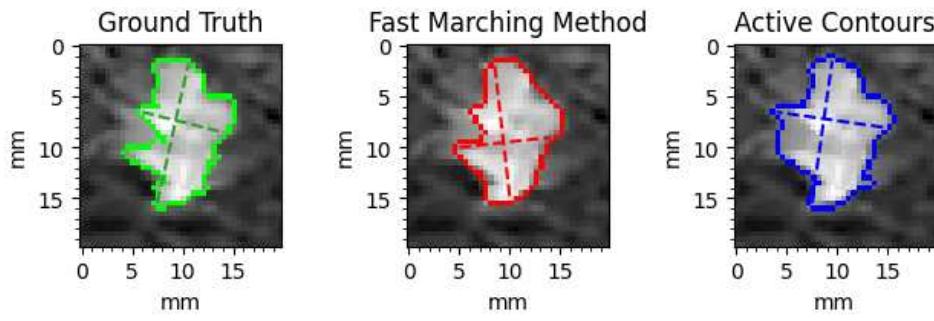
**26**

	IoU	Long-axis err.	Short-axis err.
FMM	0.732	37.1%	13.8%
AC	0.774	4.36%	10.3%



**27**

	IoU	Long-axis err.	Short-axis err.
FMM	0.858	5.35%	9.95%
AC	0.793	2.24%	28.6%



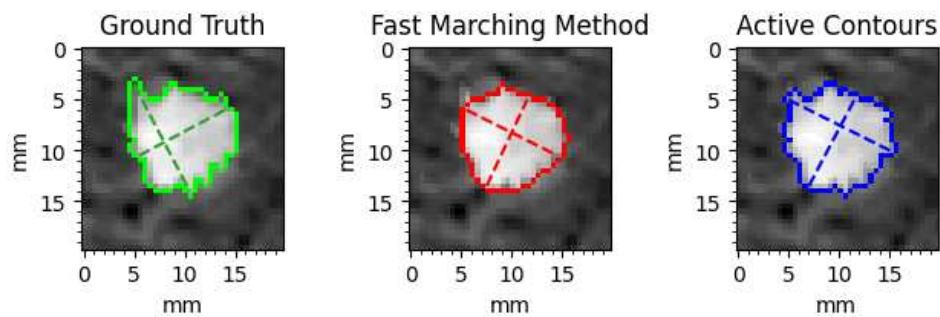
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

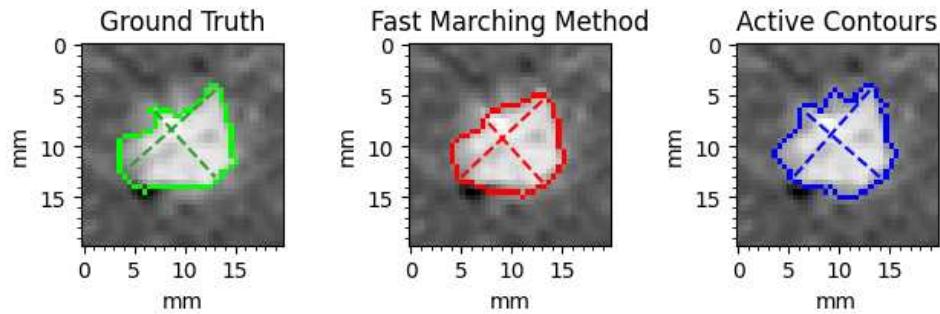
**28**

	IoU	Long-axis err.	Short-axis err.
FMM	0.860	15.1%	2.29%
AC	0.910	5.65%	0%



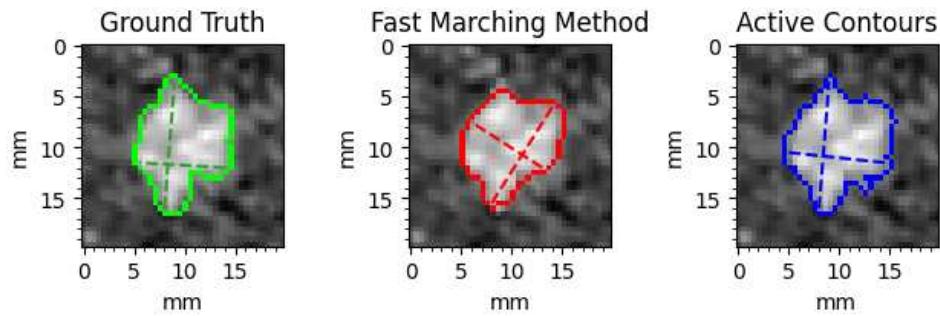
**29**

	IoU	Long-axis err.	Short-axis err.
FMM	0.844	8.89%	8.02%
AC	0.785	5.81%	12.2%



**30**

	IoU	Long-axis err.	Short-axis err.
FMM	0.824	10.4%	2.35%
AC	0.846	0%	12.1%



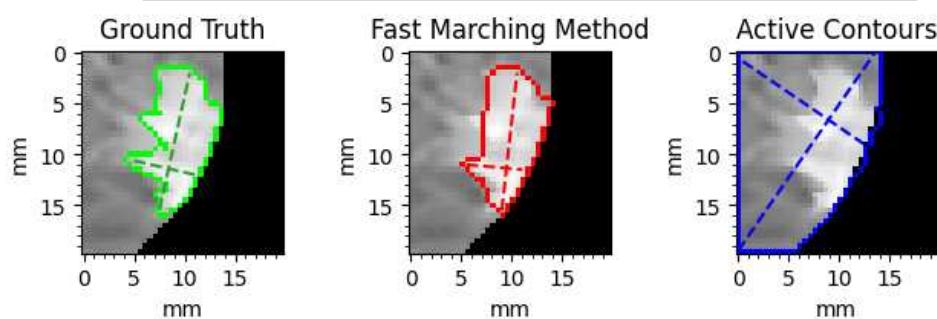
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

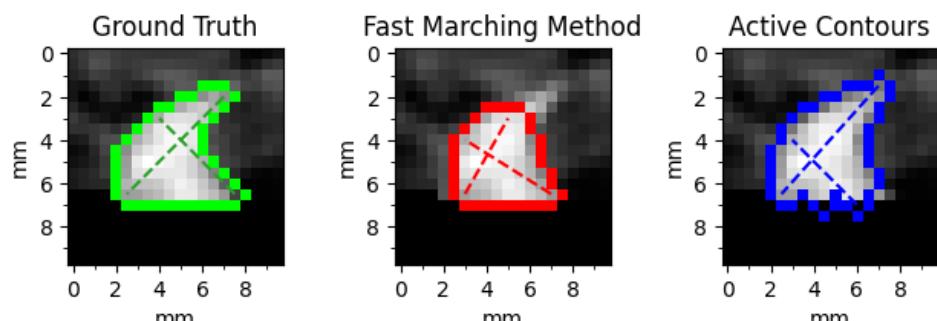
**31**

	IoU	Long-axis err.	Short-axis err.
FMM	0.847	1.81%	17.2%
AC	0.269	71.5%	127%



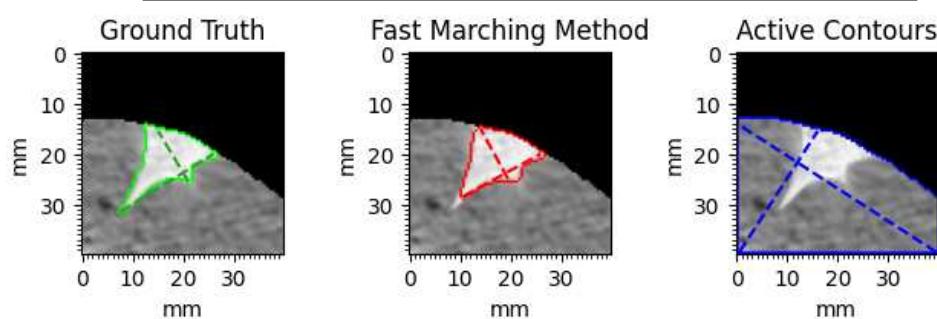
**32**

	IoU	Long-axis err.	Short-axis err.
FMM	0.757	25.8%	18.6%
AC	0.785	5.82%	14.3%



**33**

	IoU	Long-axis err.	Short-axis err.
FMM	0.868	18.8%	2.06%
AC	0.134	116%	145%



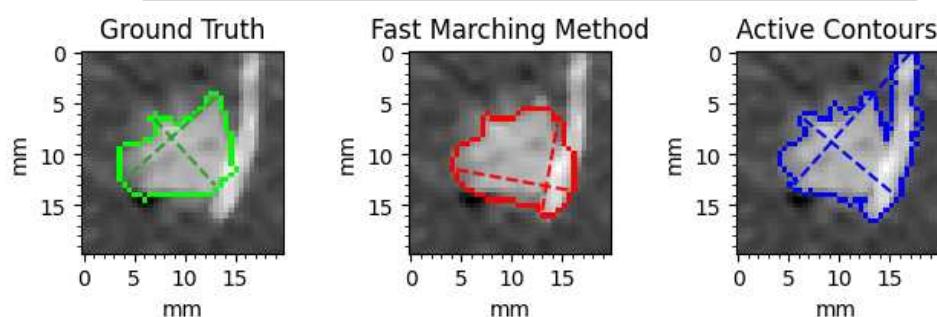
---

 APPENDIX B. SEGMENTATION RESULTS - PHANTOM
 

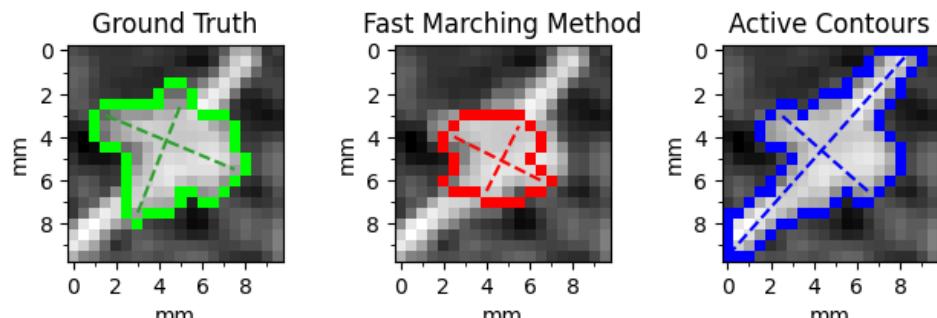
---

**34**

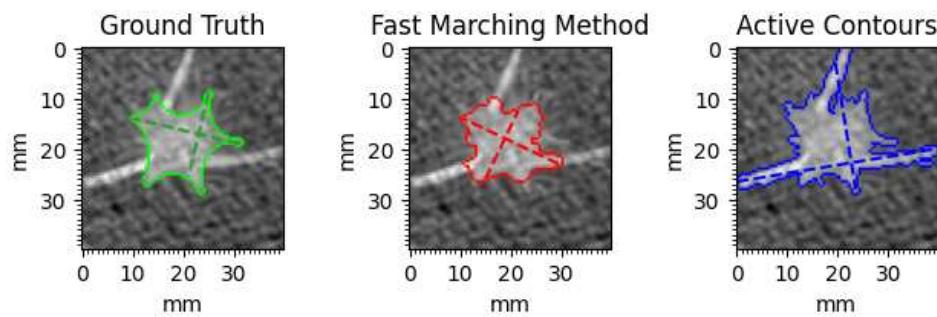
	IoU	Long-axis err.	Short-axis err.
FMM	0.706	7.14%	2.49%
AC	0.571	44.2%	32.4%


**35**

	IoU	Long-axis err.	Short-axis err.
FMM	0.500	31.2%	37.5%
AC	0.566	96.2%	1.11%


**36**

	IoU	Long-axis err.	Short-axis err.
FMM	0.732	2.64%	11.8%
AC	0.573	85.4%	49.1%



---

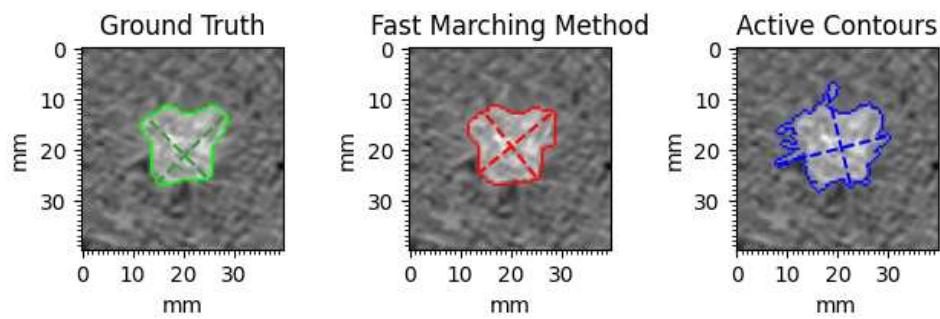
APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

### B.3 Semi-transparent

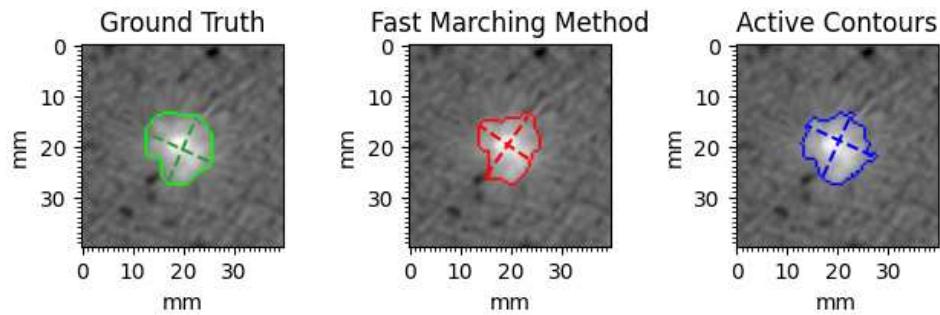
**1**

	IoU	Long-axis err.	Short-axis err.
FMM	0.903	1.42%	1.54%
AC	0.780	20.7%	14.1%



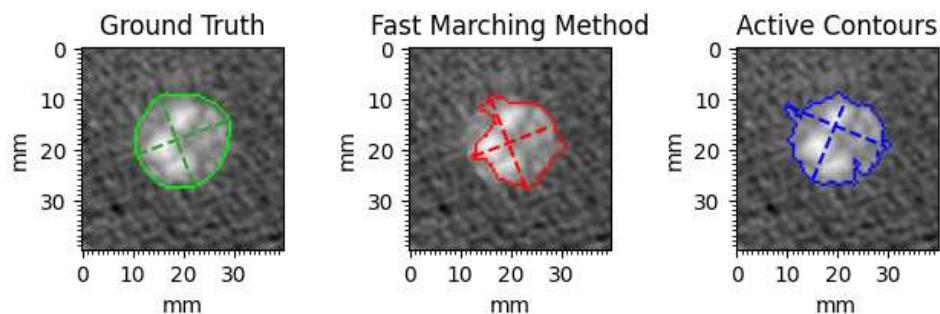
**2**

	IoU	Long-axis err.	Short-axis err.
FMM	0.790	7.69%	12.8%
AC	0.853	4.83%	0%



**3**

	IoU	Long-axis err.	Short-axis err.
FMM	0.685	0.964%	8.68%
AC	0.847	12.9%	7.64%



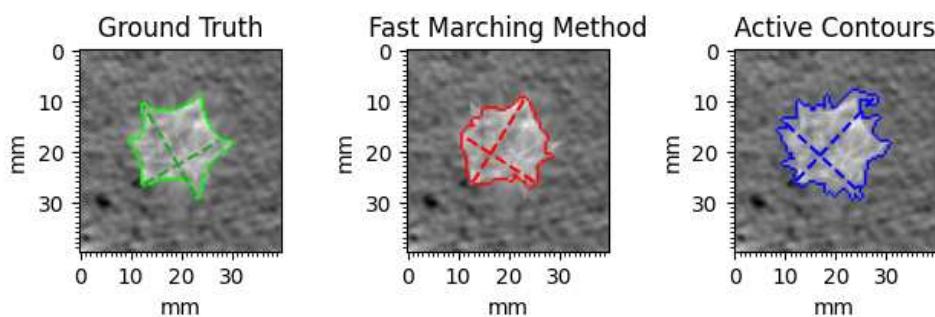
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

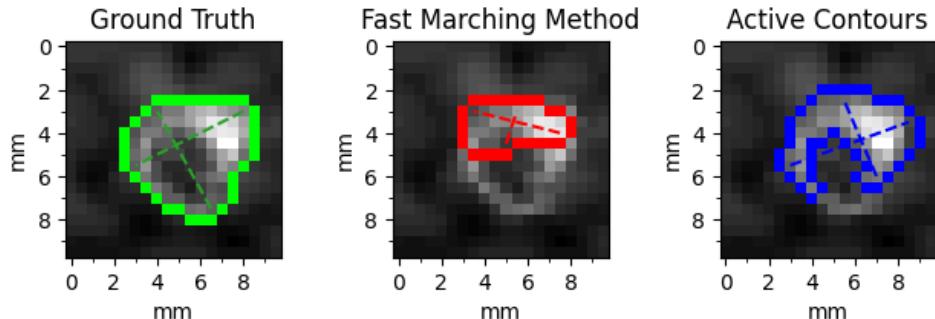
**4**

	IoU	Long-axis err.	Short-axis err.
FMM	0.836	9.45%	3.48%
AC	0.759	8.64%	14.9%



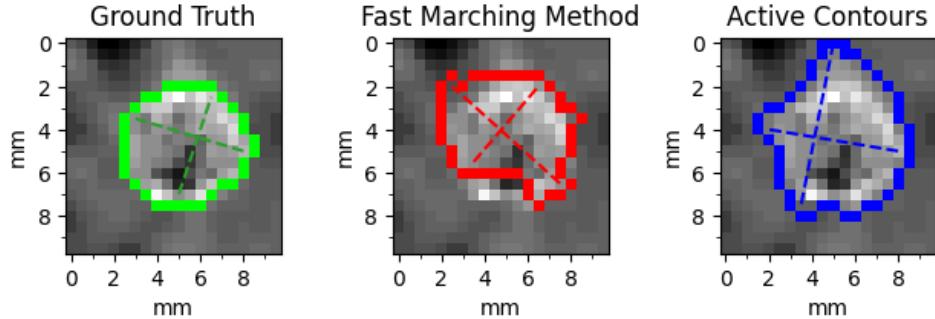
**5**

	IoU	Long-axis err.	Short-axis err.
FMM	0.322	26.3%	69.3%
AC	0.511	4.65%	26.0%



**6**

	IoU	Long-axis err.	Short-axis err.
FMM	0.606	28.9%	2.74%
AC	0.680	46.6%	28.3%



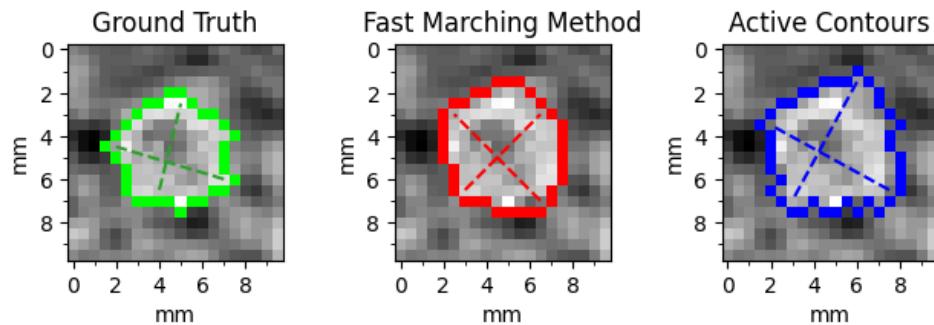
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

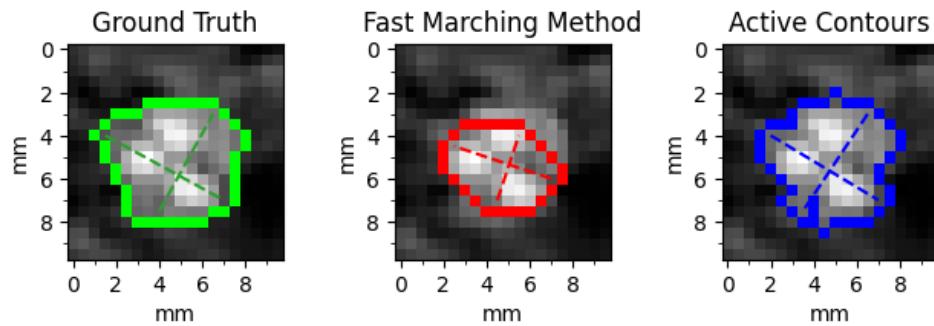
**7**

	IoU	Long-axis err.	Short-axis err.
FMM	0.756	8.43%	20.1%
AC	0.648	19.9%	51.9%



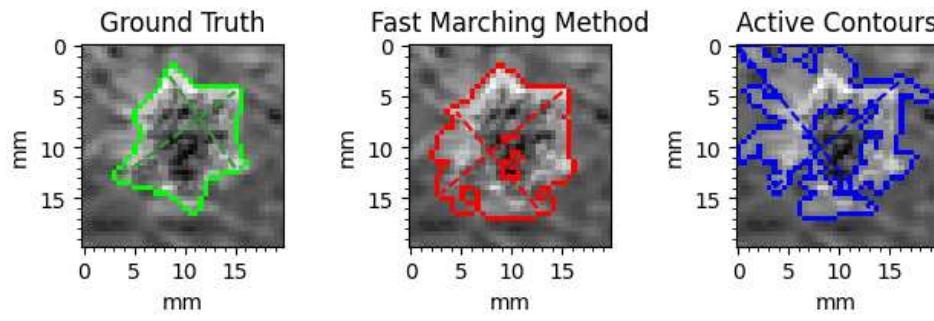
**8**

	IoU	Long-axis err.	Short-axis err.
FMM	0.545	24.3%	38.6%
AC	0.863	6.87%	5.05%



**9**

	IoU	Long-axis err.	Short-axis err.
FMM	0.686	9.23%	11.9%
AC	0.299	44.2%	47.0%



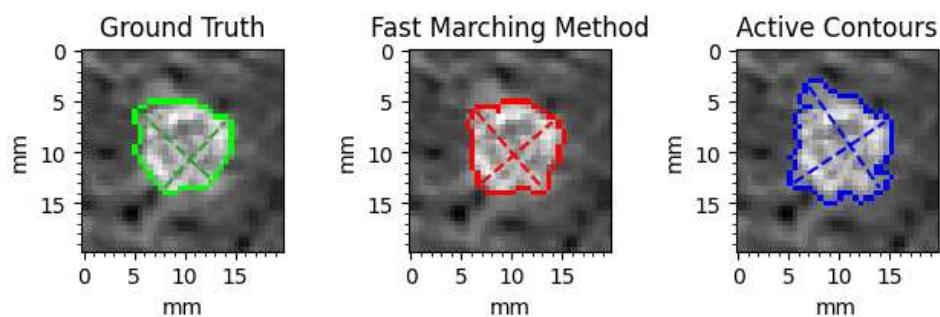
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

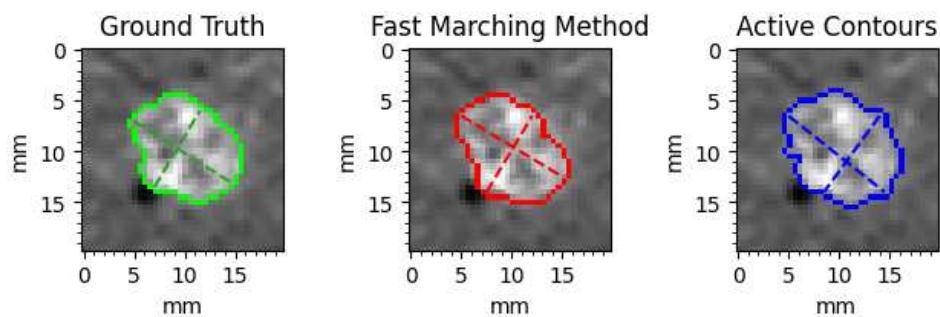
**10**

	IoU	Long-axis err.	Short-axis err.
FMM	0.848	3.87%	4.18%
AC	0.715	27.9%	22.3%



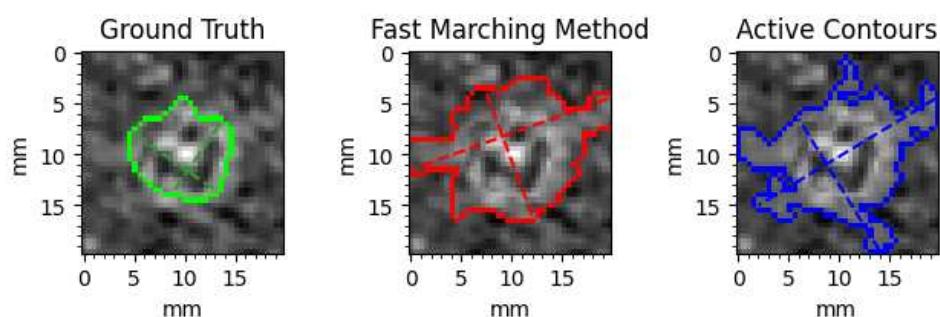
**11**

	IoU	Long-axis err.	Short-axis err.
FMM	0.868	3.74%	0%
AC	0.870	7.65%	6.41%



**12**

	IoU	Long-axis err.	Short-axis err.
FMM	0.417	113%	62.3%
AC	0.443	110%	93.2%



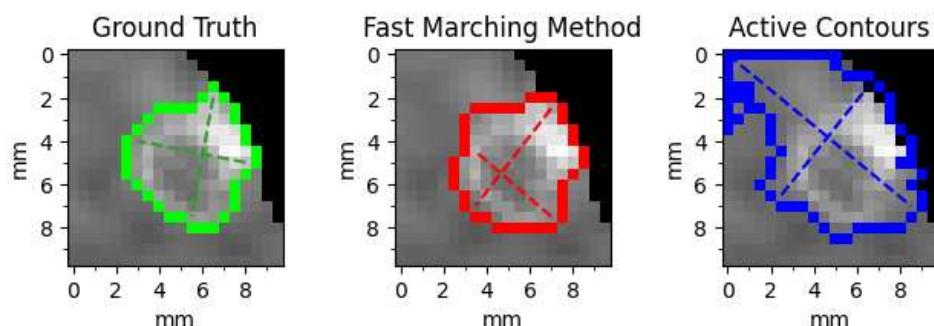
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

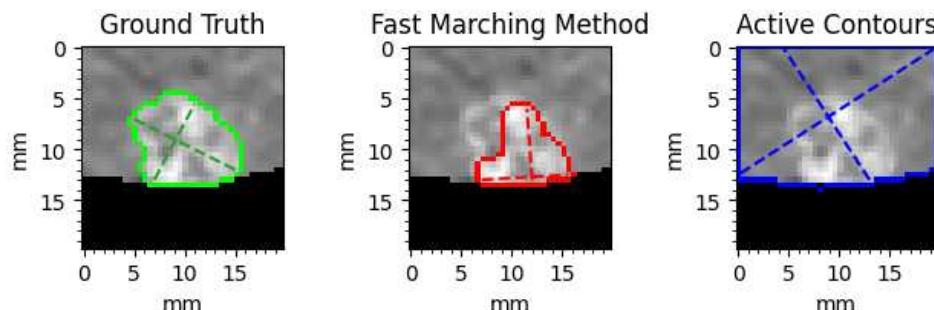
**13**

	IoU	Long-axis err.	Short-axis err.
FMM	0.883	1.97%	9.61%
AC	0.459	97.0%	25.5%



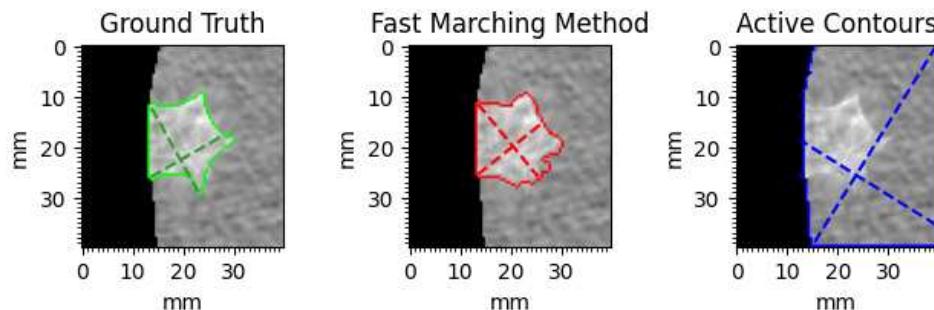
**14**

	IoU	Long-axis err.	Short-axis err.
FMM	0.568	23.9%	17.4%
AC	0.258	107%	82.7%



**15**

	IoU	Long-axis err.	Short-axis err.
FMM	0.849	3.34%	0.682%
AC	0.174	139%	93.7%



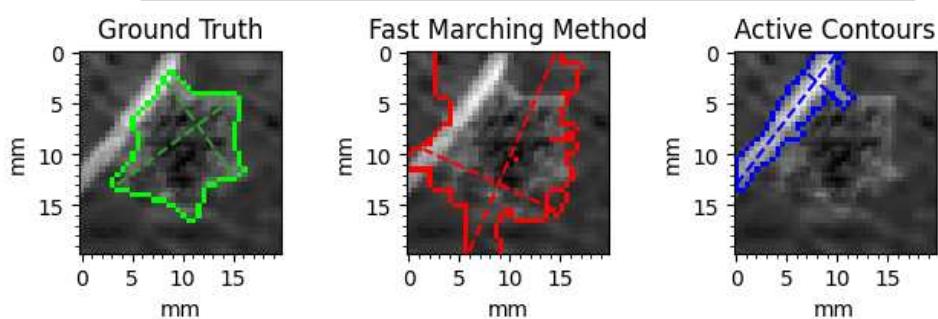
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

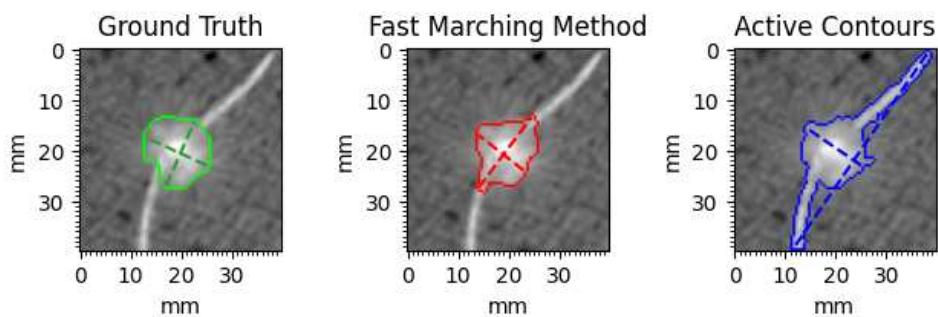
**16**

	IoU	Long-axis err.	Short-axis err.
FMM	0.468	48.7%	29.0%
AC	0.0650	14.7%	58.5%



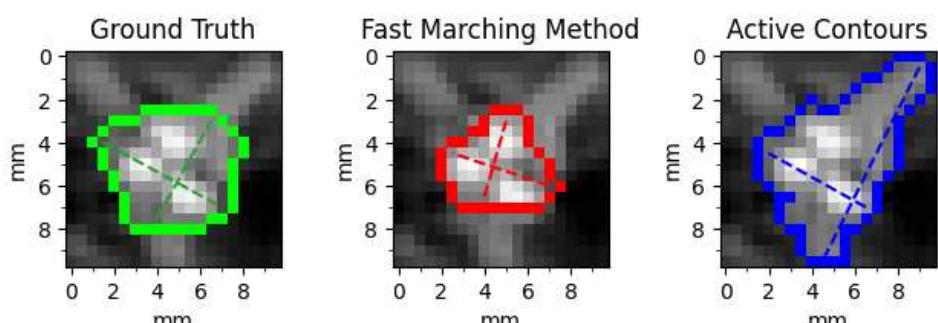
**17**

	IoU	Long-axis err.	Short-axis err.
FMM	0.747	28.1%	13.6%
AC	0.574	242%	0.909%



**18**

	IoU	Long-axis err.	Short-axis err.
FMM	0.545	24.3%	29.3%
AC	0.648	60.7%	0%



---

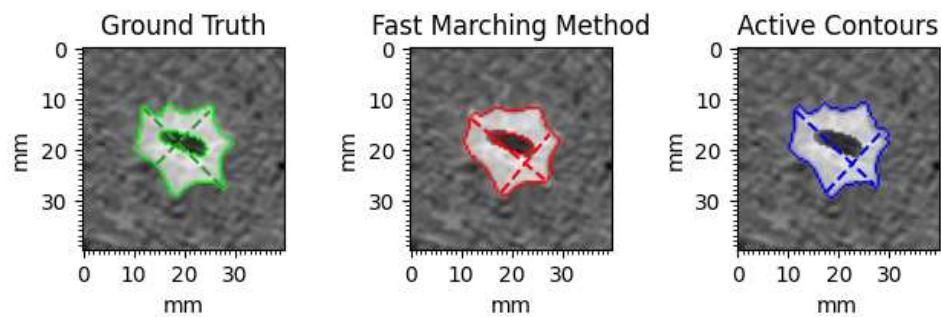
APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

#### B.4 Cavitary

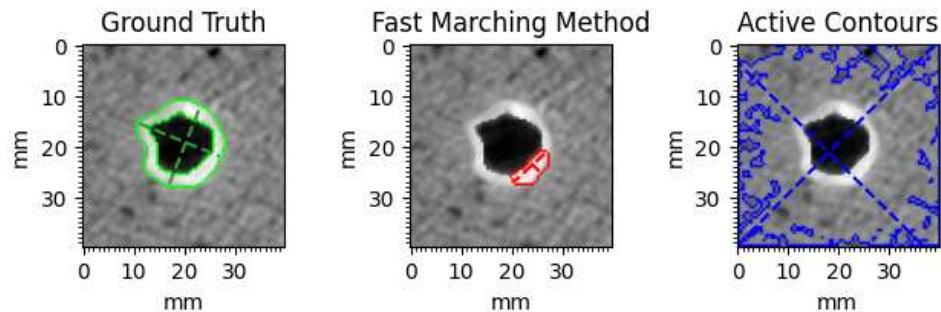
**1**

	IoU	Long-axis err.	Short-axis err.
FMM	0.862	14.1%	0.471%
AC	0.752	3.19%	9.63%



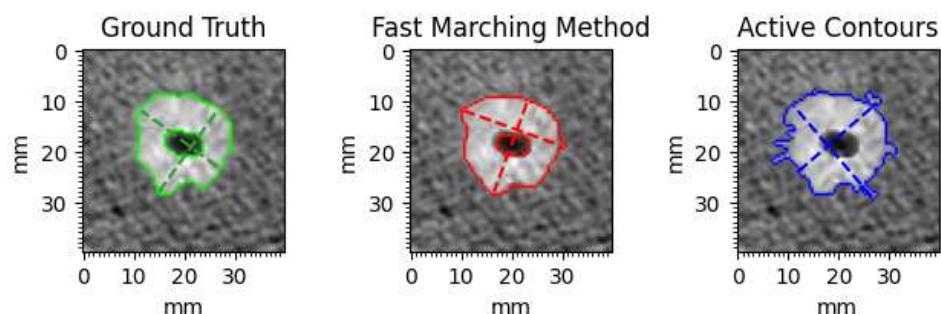
**2**

	IoU	Long-axis err.	Short-axis err.
FMM	0.191	52.9%	82.5%
AC	0.0700	221%	217%



**3**

	IoU	Long-axis err.	Short-axis err.
FMM	0.886	5.18%	0.376%
AC	0.749	20.6%	10.5%



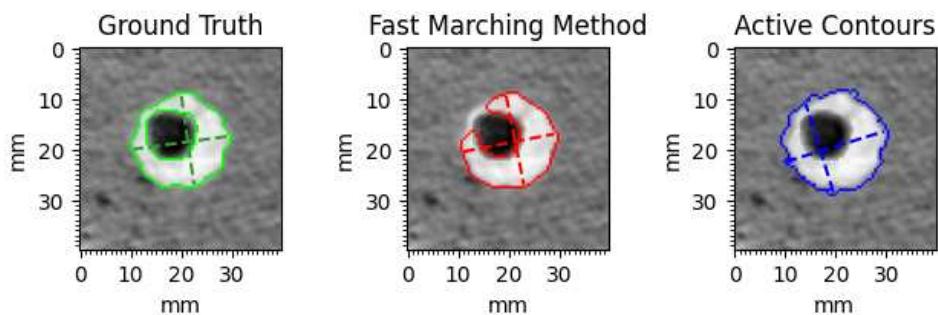
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

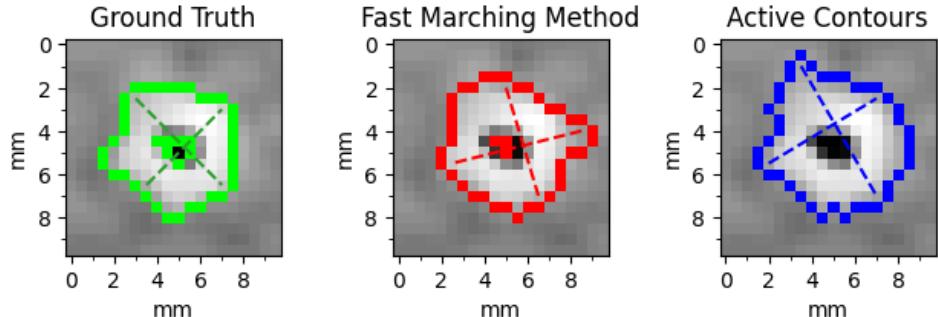
**4**

	IoU	Long-axis err.	Short-axis err.
FMM	0.796	4.39%	3.73%
AC	0.564	9.27%	3.73%



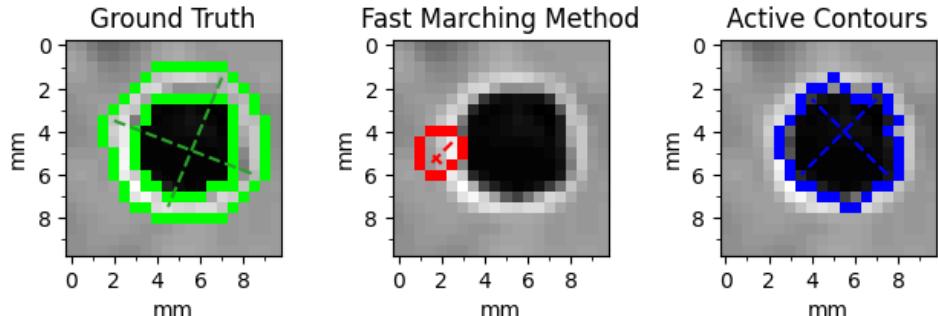
**5**

	IoU	Long-axis err.	Short-axis err.
FMM	0.740	9.19%	5.46%
AC	0.690	22.8%	17.8%



**6**

	IoU	Long-axis err.	Short-axis err.
FMM	0.0450	79.7%	89.2%
AC	0.00700	28.9%	23.8%



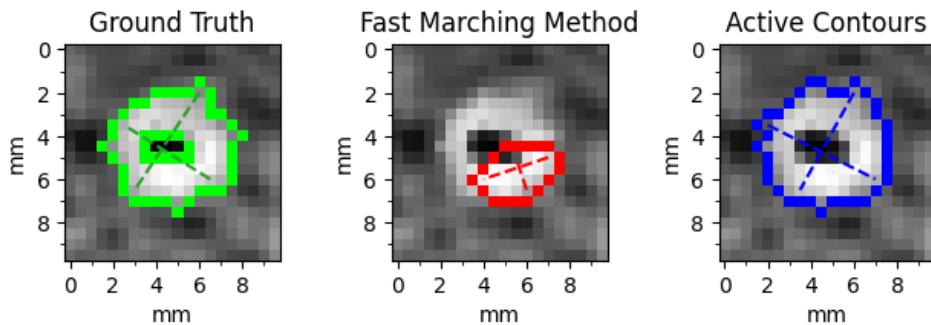
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

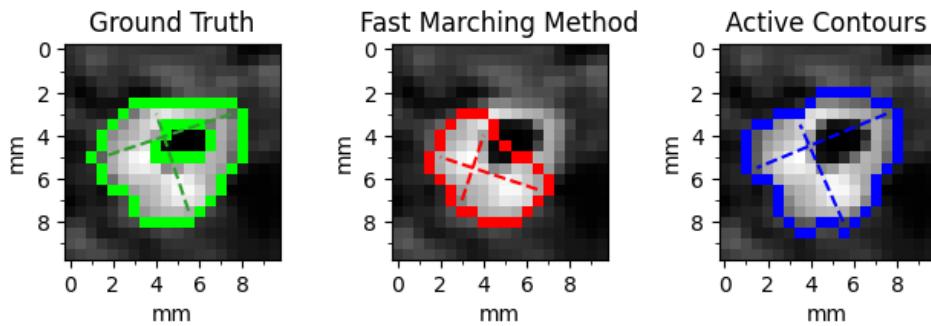
**7**

	IoU	Long-axis err.	Short-axis err.
FMM	0.261	41.6%	66.5%
AC	0.703	3.33%	9.11%



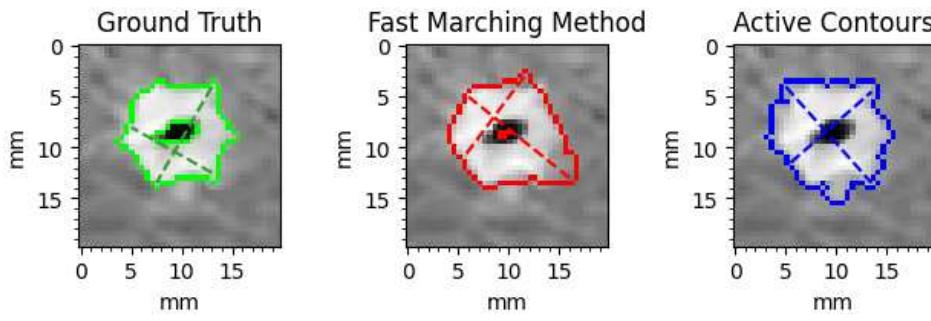
**8**

	IoU	Long-axis err.	Short-axis err.
FMM	0.595	25.0%	33.3%
AC	0.664	2.85%	3.80%



**9**

	IoU	Long-axis err.	Short-axis err.
FMM	0.706	16.7%	8.93%
AC	0.674	16.1%	15.8%



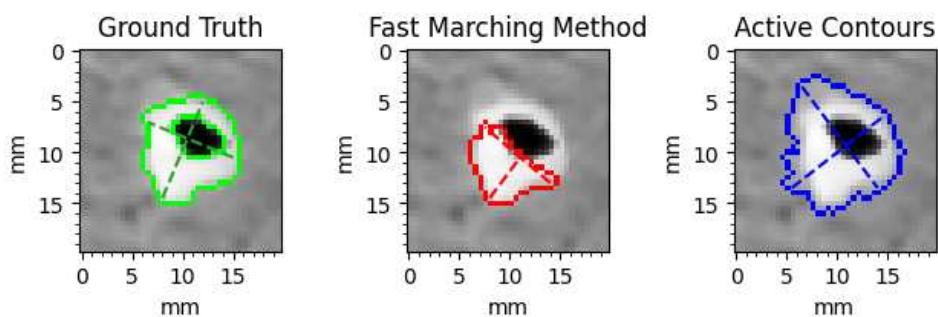
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

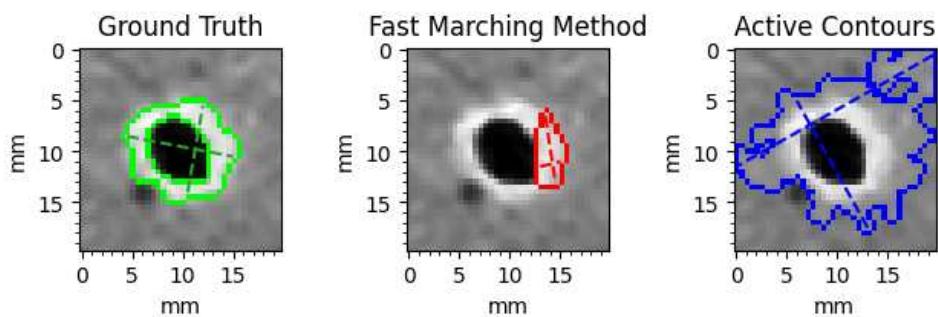
**10**

	IoU	Long-axis err.	Short-axis err.
FMM	0.521	17.5%	45.6%
AC	0.519	21.2%	24.0%



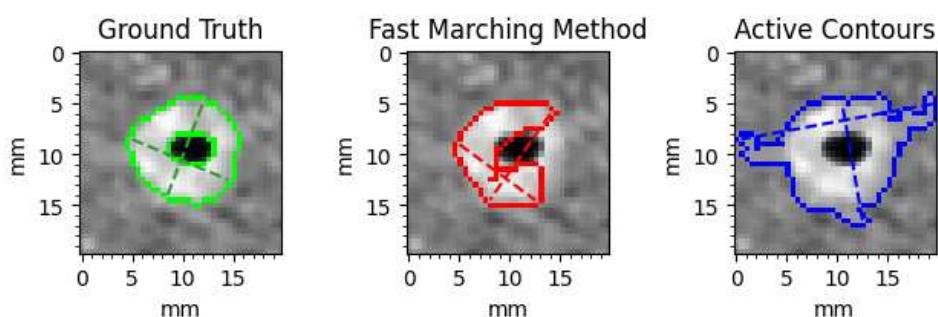
**11**

	IoU	Long-axis err.	Short-axis err.
FMM	0.239	38.4%	76.1%
AC	0.199	109%	65.9%



**12**

	IoU	Long-axis err.	Short-axis err.
FMM	0.636	0.873%	3.63%
AC	0.556	87.4%	21.0%



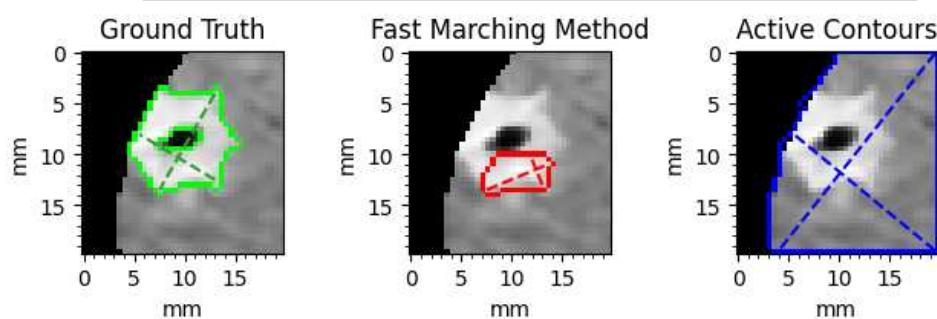
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

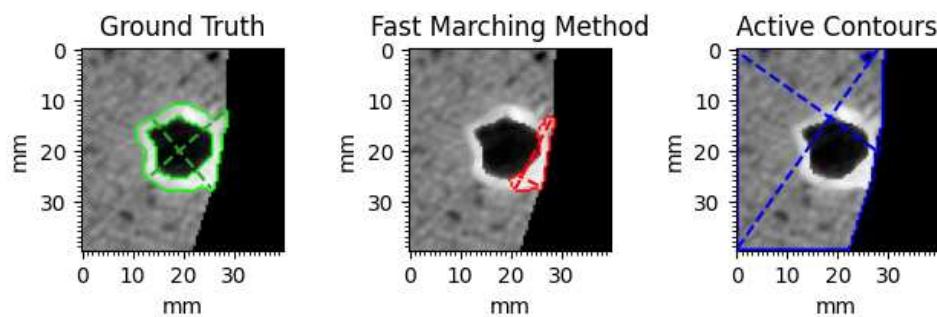
**13**

	IoU	Long-axis err.	Short-axis err.
FMM	0.243	40.8%	69.1%
AC	0.209	127%	107%



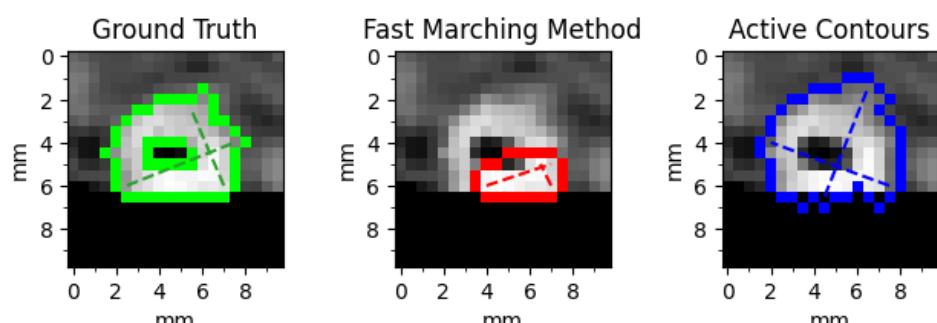
**14**

	IoU	Long-axis err.	Short-axis err.
FMM	0.342	23.4%	74.8%
AC	0.0970	144%	90.0%



**15**

	IoU	Long-axis err.	Short-axis err.
FMM	0.270	41.4%	70.6%
AC	0.606	8.53%	41.2%



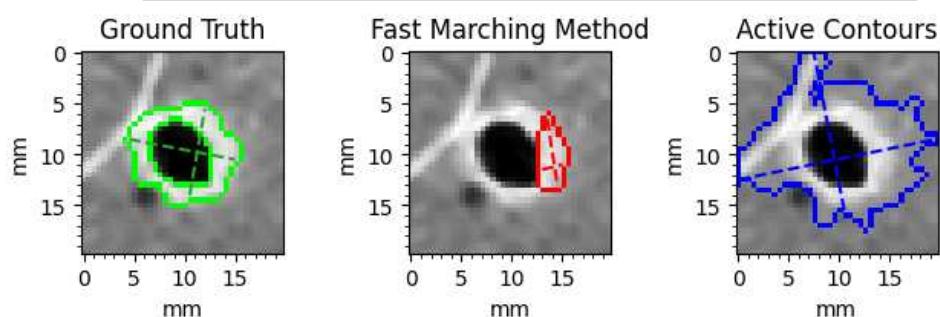
---

APPENDIX B. SEGMENTATION RESULTS - PHANTOM

---

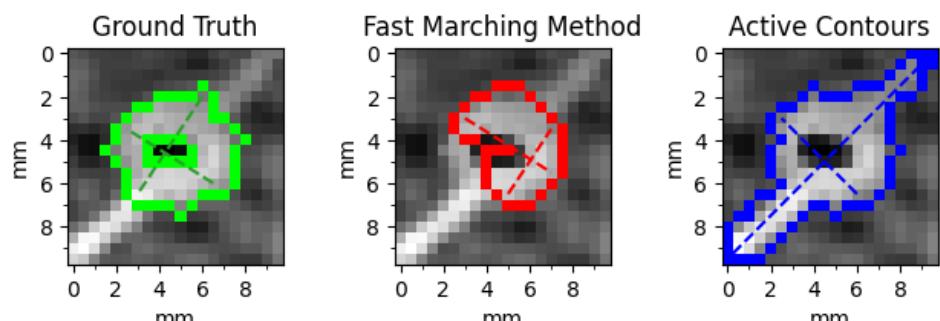
**16**

	IoU	Long-axis err.	Short-axis err.
FMM	0.239	38.4%	76.1%
AC	0.220	86.2%	83.0%



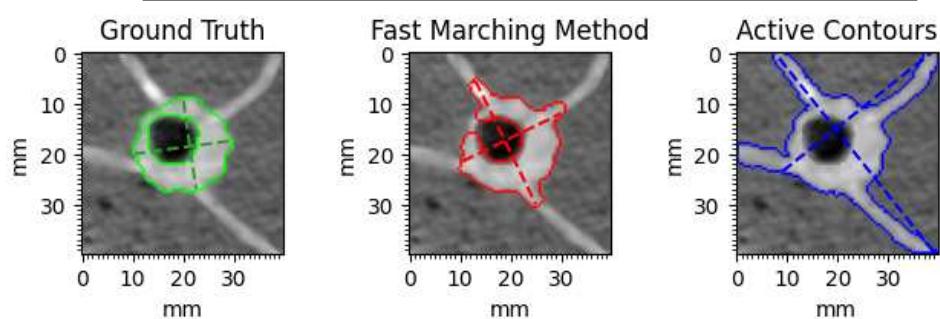
**17**

	IoU	Long-axis err.	Short-axis err.
FMM	0.557	12.8%	23.7%
AC	0.488	148%	4.87%



**18**

	IoU	Long-axis err.	Short-axis err.
FMM	0.660	46.1%	22.7%
AC	0.355	171%	113%



# Bibliography

- [1] J. Ahmedin S.R. L., M.K. D. Cancer statistics, 2017. *CA Cancer J Clin.* 67, pages 7–30, 2017.
- [2] R.T.H. Leijenaar C. Parmar P. Grossmann S. Carvalho J. Bussink R. Monshouwer et al. H.J.W.L. Aerts, E.R. Velazquez. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nat. Commun.* 5, 2014.
- [3] Adams AM et al. National Lung Screening Trial Research T, Aberle DR. Reduced lung-cancer mortality with low-dose computed tomographic screening. *N Engl J Med* 2011;365, pages 395–409, 2011.
- [4] U.S. Preventive Services Task Force. Screening for lung cancer Moyer VA. U.s. preventive services task force recommendation statement. *Ann Intern Med*, pages 160:330–8, 2014.
- [5] W. J. Kostis, A. P. Reeves, D. F. Yankelevitz, and C. I. Henschke. Three-dimensional segmentation and growth-rate estimation of small pulmonary nodules in helical ct images. *IEEE Transactions on Medical Imaging*, 22(10):1259–1274, 2003.
- [6] J. . Kuhnigk, V. Dicken, L. Bornemann, A. Bakai, D. Wormanns, S. Krass, and H. . Peitgen. Morphological segmentation and partial volume analysis for volumetry of solid pulmonary lesions in thoracic ct scans. *IEEE Transactions on Medical Imaging*, 25(4):417–434, 2006.
- [7] Toshiro Kubota, Anna K. Jerebko, Maneesh Dewan, Marcos Salganicoff, and Arun Krishnan. Segmentation of pulmonary nodules of various densities with morphological approaches and convexity models. *Medical Image Analysis*, 15(1):133 – 154, 2011.
- [8] J. Dehmeshki, H. Amin, M. Valdivieso, and X. Ye. Segmentation of pulmonary nodules in thoracic ct scans: A region growing approach. *IEEE Transactions on Medical Imaging*, 27(4):467–480, 2008.
- [9] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

---

## BIBLIOGRAPHY

---

- [10] A. A. Farag, H. E. A. E. Munim, J. H. Graham, and A. A. Farag. A novel approach for lung nodules segmentation in chest ct using level sets. *IEEE Transactions on Image Processing*, 22(12):5202–5213, 2013.
- [11] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124—1137, September 2004.
- [12] Weiwei Du Yanhe Ma Yanli Miao, Jianming Wang and Hong Zhang. Volumetric measurement of ground-glass opacity nodules using expectation-maximization algorithm. *The 4th IIAE International Conference on Intelligent Systems and Image Processing 2016: The Institute of Industrial Applications Engineers. Japan: Kyoto*, pages 317–321, 2016.
- [13] W. Du Y. Ma Y. Miao, J. Wang and H. Zhang. Feature extraction of ground-glass opacity nodules using active contour model for lung cancer detection. *Advances in Computer Science Reaserach (ACSR)*, ISSN 2352-538x, Vol. 71, pages 1312–1317, 2017.
- [14] L. Lu, J. Bi, M. Wolf, and M. Salganicoff. Effective 3d object detection and regression using probabilistic segmentation features in ct images. In *CVPR 2011*, pages 1049–1056, 2011.
- [15] Yaoxiu Hu and Prahlad Menon. A neural network approach to lung nodule segmentation. page 97842O, 03 2016.
- [16] Luís Gonçalves, Jorge Novo, and Aurélio Campilho. Hessian based approaches for 3d lung nodule segmentation. *Expert Systems with Applications*, 61, 05 2016.
- [17] Julip Jung, Helen Hong, and Jin Mo Goo. Ground-glass nodule segmentation in chest ct images using asymmetric multi-phase deformable model and pulmonary vessel removal. *Computers in Biology and Medicine*, 92, 11 2017.
- [18] S. Wang, M. Zhou, O. Gevaert, Z. Tang, D. Dong, Z. Liu, and T. Jie. A multi-view deep convolutional neural networks for lung nodule segmentation. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1752–1755, 2017.
- [19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

---

## BIBLIOGRAPHY

---

- [20] S. Wang, M. Zhou, Zaiyi Liu, Zhenyu Liu, Dongsheng Gu, Y. Zang, D. Dong, O. Gevaert, and J. Tian. Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation. *Medical image analysis*, 40:172 – 183, 2017.
- [21] Haichao Cao, Hong Liu, Enmin Song, Chih-Cheng Hung, Guangzhi Ma, Xiangyang Xu, Renchao Jin, and Jianguo Lu. Dual-branch residual network for lung nodule segmentation. *Applied Soft Computing*, 86:105934, 2020.
- [22] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.
- [23] Guido Zuidhof. *Full Preprocessing Tutorial*, 2016).
- [24] Arnav Jain. *Candidate Generation and LUNA16 preprocessing*, 2017).
- [25] Ziv Yaniv, Bradley Lowekamp, Hans Johnson, and Richard Beare. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of Digital Imaging*, 31, 11 2017.
- [26] J. Hearn. Competitive medical image segmentation with the fast marching method. 2008.
- [27] J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [28] Eftichis Sifakis, Christophe Garcia, and Georgios Tziritas. Bayesian level sets for image segmentation. 13(1):44–64, March 2002.
- [29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [30] MATLAB. *version 9.8 (R2020a)*. The MathWorks Inc., Natick, Massachusetts, 2020.
- [31] Inc Phaseit. *PyPDF2 - The PdfFileReader Class*, 2016.
- [32] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *The British Journal of Radiology*, 46(552):1016–1022, 1973. PMID: 4757352.
- [33] Conall J Garvey and Rebecca Hanlon. Computed tomography in clinical practice. *BMJ*, 324(7345):1077–1080, 2002.

---

## BIBLIOGRAPHY

---

- [34] Siva P. Raman, Mahadevappa Mahesh, Robert V. Blasko, and Elliot K. Fishman. Ct scan parameters and radiation dose: Practical advice for radiologists. *Journal of the American College of Radiology*, 10(11):840–846, November 2013.
- [35] Traverso A. De Bel T. et al Setio, A. A .A. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The luna16 challenge[j]. *Medical Image Analysis*, pages 42:1–13, 2017.
- [36] Apanasovich TV et al. Reeves AP, Biancardi AM. The lung image database consortium (lidc): a comparison of different size metrics for pulmonary nodule measurements. *Acad Radiol.* 14, page 1475–1485, 2007.
- [37] Darcy Mason. Su-e-t-33: pydicom: an open source dicom library. *Medical Physics*, 38(6Part10):3493–3493, 2011.
- [38] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [39] G. N. Hounsfield. Computed medical imaging. Nobel lecture, December 8, 1979. *J Comput Assist Tomogr*, 4(5):665–674, Oct 1980.
- [40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Paul van der Walt, et al., and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.
- [41] Butterworth Stephen. On the theory of filter amplifiers. *Wireless Engineer (also called Experimental Wireless and the Wireless Engineer)*, 7:536–541, 1930.
- [42] McGillem D. Clare; Cooper R. George. *Continuous and Discrete Signal and System Analysis (2 ed.)*. Holt, Rinehart and Winston, 1984.
- [43] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuel Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.
- [44] Nicolas Forcadel, Carole Guyader, and Christian Gout. Generalized fast marching method: Applications to image segmentation. *Numerical Algorithms*, 48:189–211, 07 2008.
- [45] X. Song, M. Cheng, B. Wang, S. Huang, and X. Huang. Automatic liver segmentation from ct images using adaptive fast marching method. In

---

## BIBLIOGRAPHY

---

- 2013 *Seventh International Conference on Image and Graphics*, pages 897–900, 2013.
- [46] Z. Clawson, A. Chacon, and A. Vladimirsny. Causal domain restriction for eikonal equations. *SIAM Journal on Scientific Computing*, 36(5):A2478–A2505, 2014.
  - [47] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
  - [48] Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
  - [49] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29(3):867–884, June 1992.
  - [50] Ron Kimmel. Fast marching methods for computing distance maps and shortest paths. 1996.
  - [51] J.A. Sethian. Fast marching methods and level set methods for propagating interfaces. 1998.
  - [52] Bradley Lowekamp, David Chen, Luis Ibanez, and Daniel Blezek. The design of simpleitk. *Frontiers in Neuroinformatics*, 7:45, 2013.
  - [53] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277 – 1294, 1993.
  - [54] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
  - [55] Ross T. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, Sep 1998.
  - [56] S. G. Armato, G. McLennan, and L. et al Bidaut. The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): a completed reference database of lung nodules on CT scans. *Med Phys*, 38(2):915–931, Feb 2011.
  - [57] Alberto M Biancardi, Artit C Jirapatnakul, and Anthony P Reeves. A comparison of ground truth estimation methods. *International journal of computer assisted radiology and surgery*, 5(3):295—305, May 2010.
  - [58] Hantao Liu and Zhou Wang. *Perceptual Quality Assessment of Medical Images*. 01 2017.

---

## BIBLIOGRAPHY

---

- [59] Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. A subjective experiment for 3d-mesh segmentation evaluation. pages 356 – 360, 11 2010.
- [60] Konstantinos Loverdos, Andreas Fotiadis, Chrysoula Kontogianni, Marianthi Iliopoulou, and Mina Gaga. Lung nodules: A comprehensive review on current approach and management. *Annals of Thoracic Medicine*, 14:226, 10 2019.
- [61] David M. Hansell, Alexander A. Bankier, Heber MacMahon, Theresa C. McLoud, Nestor L. Müller, and Jacques Remy. Fleischner society: Glossary of terms for thoracic imaging. *Radiology*, 246(3):697–722, 2008. PMID: 18195376.
- [62] The GIMP Development Team. Gimp.
- [63] Alexander A. Bankier, Heber MacMahon, Jin Mo Goo, Geoffrey D. Rubin, Cornelia M. Schaefer-Prokop, and David P. Naidich. Recommendations for measuring pulmonary nodules at ct: A statement from the fleischner society. *Radiology*, 285(2):584–600, 2017. PMID: 28650738.
- [64] Kelly Zou, Simon Warfield, Aditya Bharatha, Clare Tempany, Michael Kaus, Steven Haker, William Wells, Ferenc Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic radiology*, 11:178–89, 02 2004.
- [65] Joon Lee Chang-Mo, NamJihang KimKyong. Lung nodule segmentation with convolutional neuralnetwork trained by simple diameter information. *1st Conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam, The Netherlands*.
- [66] Marjolein A. Heuvelmans Daiwei Han and Matthijs Oudkerk. Volume versus diameter assessment of small pulmonary nodules in ct lung cancer screening. *Transl Lung Cancer Res.*, page 6(1): 52–61, 2017.
- [67] S. Diciotti, S. Lombardo, M. Falchini, G. Picozzi, and M. Mascalchi. Automated segmentation refinement of small lung nodules in ct scans by local shape analysis. *IEEE Transactions on Biomedical Engineering*, 58(12):3418–3428, 2011.
- [68] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.