A Segmentation Method for Stentgrafts in the Abdominal Aorta from ECG-gated CTA Data

Almar Klein[a], W. KlaasJan Renema[b], Luuk J. Oostveen[b], Leo J. Schultze Kool[b], Cornelis H. Slump[a]

[a]Signals and Systems group, University of Twente, Enschede, The Netherlands;
[b]Dept. of Radiology, Radboud University Nijmegen Medical Centre, Nijmegen, The Netherlands

ABSTRACT

Endovascular aortic replacement (EVAR) is an established technique, which uses stentgrafts to treat aortic aneurysms in patients at risk of aneurysm rupture. The long-term durability of a stentgraft is affected by the stresses and hemodynamic forces applied to it, and may be reflected by the movements of the stentgraft itself during the cardiac cycle. A conventional CT scan (which results in a 3D volume) is not able to visualize these movements. However, applying ECG-gating does provide insight in the motion of the stentgraft caused by hemodynamic forces at different phases of the cardiac cycle.

The amount of data obtained is a factor of ten larger compared to conventional CT, but the radiation dose is kept similar for patient safety. This causes the data to be noisy, and streak artifacts are more common. Algorithms for automatic stentgraft detection must be able to cope with this.

Segmentation of the stentgraft is performed by examining slices perpendicular to the centreline. Regions with high CT-values exist at the locations where the metallic frame penetrates the slice. These regions are well suited for detection and sub-pixel localization. Spurious points can be removed by means of a clustering algorithm, leaving only points on the contour of the stent. We compare the performance of several different point detection methods and clustering algorithms. The position of the stent's centreline is calculated by fitting a circle through these points.

The proposed method can detect several stentgraft types, and is robust against noise and streak artifacts.

Keywords: cardiovascular biomechanics and modeling, image-based biomechanical models, segmentation, motion analysis, EVAR, CT, stentgraft

1. INTRODUCTION

Endovascular Aortic replacement (EVAR) for the treatment of aortic aneurysms in patients at risk of aneurysm rupture is an established endovascular technique.[1] However, the development of the stentgraft design is ongoing in order to prevent late stentgraft failure (e.g. graft breakage, graft displacement). The stresses applied to the stentgraft by the high physiological forces and stresses in the aorta have an effect on the durability and functioning of the stentgraft. The stresses and forces that occur during movement may be reflected by the motion of the stentgraft itself during the cardiac cycle. Knowing the intrinsic properties of the stentgraft, it should be possible to evaluate these stresses and forces by calculating its displacement and deformation.

We apply a cardiac CT technique, ECG-gated CTA (computed tomography angiography), to measure the stentgraft's movements during the cardiac cycle. With a normal CT scan, a 3D volume is imaged. This new method of applying ECG-gating results in ten CT volumes, which each represent a different phase of the cardiac cycle, allowing 4D visualization of the stentgraft.

1.1 Purpose

The purpose of our research is to gain more insight in the dynamic behaviour of the stentgraft in vivo. Knowledge about the stresses acting on the stentgraft will help understand the functionality and failure of different types of stentgraft over the long term. These insights can help in stent design. Furthermore, the information on the behaviour (during the cardiac cycle) and the long term functioning of different types of stentgrafts can aid the treating physician in deciding which stent-type to use for a specific patient.

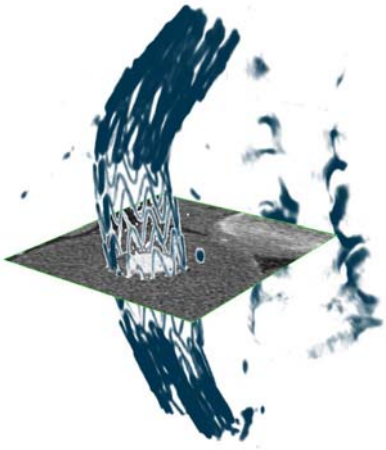


Figure 1 – Volume rendering of a stent and a plane perpendicular to the stent's centreline.

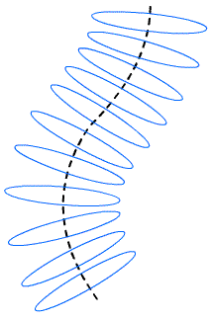


Figure 2 – Illustration of contours perpendicular to a centreline.

1.2 Problem Statement

The movements of a stentgraft can only be measured if we know its exact position. Therefore, we must first accurately segment the stentgraft. One of the challenges of ECG-gated CTA is analysing the large dataset acquired: in the order of 4000 image slices per patient. Automatic detection of the metal frame of the stentgraft will improve and speed up data processing.

Unfortunately, a simple thresholding algorithm is not sufficient for detecting the stentgraft. The first reason is that the data is relatively noisy. For patient safety, the total dose is kept similar to a normal CT scan. However, each volume is reconstructed from the fraction of the total dataset related to the specific phase of the cardiac cycle. Therefore the dose per volume is less, resulting in a lower SNR. Second, streak artifacts occur where the stentgraft's metal frame is thick or where a coil is present next to the stentgraft; due to the lower SNR, their effect is larger than in regular CT. Third, contrast agent injected in the blood results in CT-values close to the range of CT-values seen for most stentgrafts. Fourth, different stentgraft types are made from different materials (with different absorption coefficients) and differ in the thickness of their frames. In some images, stentgrafts are seen to have CT-values as low as 300 HU (Hounsfield Units).

1.3 Approach

A stent* has a tubular structure, sometimes with branches and can be modelled by a series of stacked contours which are orthogonal to the centreline (see figure 1 and 2).[2] Our approach is to segment the stent in images sampled perpendicular to its centreline, as shown in figure 1. Regions with high CT-values (typically above 500 HU) exist where the metallic frame of the stentgraft penetrates the image. These regions have high CT-values and are "pointy" (see figure 3), which makes them well suited for point detection. By segmenting the stent's contours and iteratively estimating its centreline the stent can be tracked, similar to how vessels are tracked in a recent paper by Lee et al.[3] An advantage of this 2D approach is that visualization and algorithm design is easier in 2D than it is in 3D. In this document, we discuss the process of finding the best way to detect the contour of the stent in images sampled perpendicular to its centreline.

1.4 Overview

In our research we have considered various methods and combinations thereof. In section 2 we will discuss the different methods. In section 3 we compare the methods with an experiment to be able to select the best method.

*A stentgraft consists of a metal frame surrounded by blood-proof material (the graft). When we refer to *stent*, we refer to the metallic frame: the graft is not visible on a CT scan.

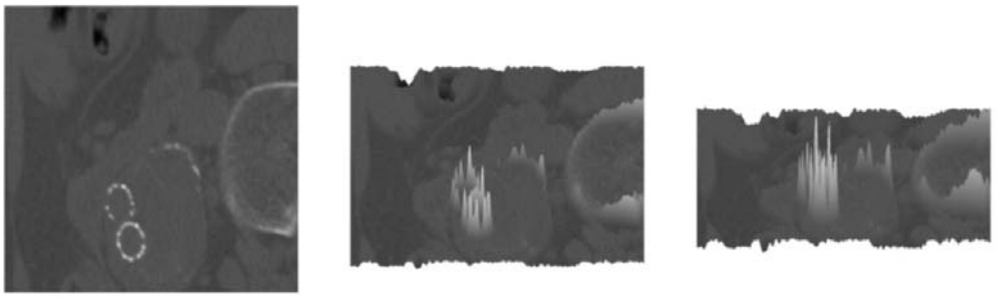


Figure 3 – The surface plot of a slice shown at different angles to illustrate the pointy structures that form the contour of the stent.

2. METHODS

In this paper we focus on segmenting the stent in a slice perpendicular to its centreline. Our approach consists of two tasks:

- Detect points on the contour of the stent.
- Find the cluster of points that really belong to the stent, removing spurious points.

In this paper we will consider four point detection methods and three clustering methods. Many of these methods use algorithms for specific basic tasks. These basic algorithms are first discussed in subsection 2.1. In the six subsections after that we will discuss the following point detection and clustering methods:

Point detection methods		**Clustering methods**	
static threshold	(2.2)	fitting	(2.5)
dynamic threshold	(2.3)	snake	(2.6)
Laplacian curvature	(2.4)	chopstick	(2.7)
Gaussian curvature	(2.4)		

2.1 Basic Algorithms

In this subsection we discuss the algorithms that are used as part of the methods described in this section.

2.1.1 Removal of metal artifacts

Metal artifacts are characterized by streaks of alternating high and low CT-values. The high streaks can cause false points to be detected. The low streaks, on the other hand, are often below 0 HU. Physiologically, tissue with negative CT-values is very unlikely to occur in the local environment of the abdominal aorta because it suggest tissue less dense than water.

We remove spurious points due to streak artifacts in all point detection methods discussed in this paper. For example in the methods of 2.3 and 2.2 such points can be found by checking for abnormally low CT-values in one of the direct neighbours of a pixel under examination.

2.1.2 Circle Fit Algorithm

The stent is designed to have the shape of a blood vessel and is thus approximately circular. Because the stent is easily deformed, however, the stent is compressed in a shape that is in general *not* a perfect circle at all. By fitting the data with a perfect circle, we generate a robust measure for the centre and radius of the stent, even when the points are not distributed in a full circle. This provides a valuable tool that is used by all of our clustering methods.

We have implemented the Modified Least Squares algorithm of Umbach and Jones.[4] Their method is a fast closed form solution and performs well even with noisy data or when only points on part of the circle are present. For details on the algorithm, we refer to their paper.[4]

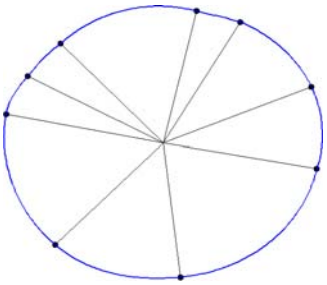


Figure 4 – Illustration of the way contour points define the circular spline.

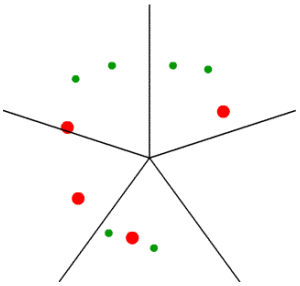


Figure 5 – Illustration of the working of the centre estimate convergence algorithm.

2.1.3 The Circular Spline

The clustering methods find points that lie on the contour of the stent. To produce a more continuous contour on which we can perform calculations, intermediate points are interpolated. We choose to interpolate in the polar coordinate system. This will produce a smooth circular shape even when large gaps are present. The centre point required to convert the points (x, y) to (r, ϕ) is estimated using the circle fit algorithm of 2.1.2. In the polar coordinate system, we use bicubic interpolation (using a cardinal spline[5] with tension $-\frac{1}{4}$) to connect the points. Figure 4 illustrates the results after the spline is converted back to Euclidean space.

2.1.4 Centre Estimate Convergence Algorithm

Given an initial estimate of the centre of the stent[†], and the set of points found by the detection method, this algorithm calculates a better estimate of the centre and the radius. The initial estimate can be near the edge of the contour. It is used by all clustering methods discussed in this paper.

The algorithm first divides the image in five segments emanating from the centre estimate (see figure 5). Next, for each segment (or direction), the closest point to the centre is selected. Then the algorithm discards points that are too far away from the centre (two times the distance of the third closest selected point). The selected set thus has a maximum of five points, which are distributed in multiple directions. Finally, the circle fit algorithm is used to estimate the centre. This process is repeated until the centre position does not change. By using five directions the algorithm will succeed (find at least three points) with gaps of over 180 degrees.

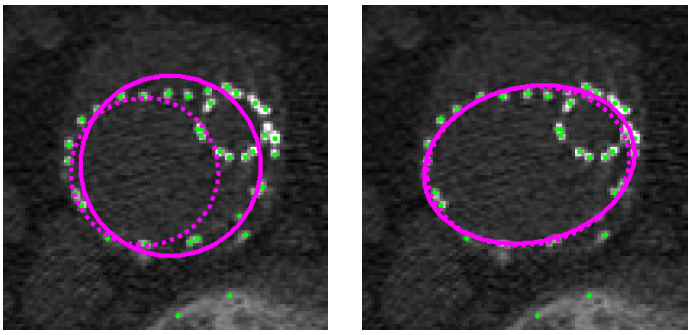
2.2 Point Detection: Static Threshold

The simplest method to detect points is to detect local maxima: points which have a larger pixelvalue than their eight neighbours. A threshold will prevent points to be found in noise. This threshold is the main parameter of this method and should be low enough to detect most points, but high enough to prevent too many spurious points, for instance in contrast fluids. The threshold was empirically determined at 650HU.

2.3 Point Detection: Dynamic Threshold

This method too, detects local maxima and requires pixelvalues to exceed a certain threshold. In this case however, the threshold is dynamic: it depends on the local environment of the pixel under examination. The threshold is applied to the pixelvalue subtracted by the average of a local patch around it. The threshold for this method is set at 260HU, which is—as expected—lower than for static thresholding. This method allows us to find parts of the stentgraft that make a small contribution to the pixel intensity, while minimizing the amount of spurious points, especially in scans where contrast fluid was used.

[†]This centre is now manually selected, but will later be calculated from the estimated centreline found in previous slices.



(a) initial and converged circle (b) initial and converged ellipse

Figure 6 – Illustration of the fitting method applied to an example.

2.4 Point Detection: Hessian Based

Differential geometry (in image processing) means extracting information (structure) using the derivatives of images.[6] As can be seen in figure 3, the spots where the stent penetrates the slice are very intense, but what is even more distinguishing is their pointy structure. This aspect can be detected using second order derivatives. The Hessian matrix is an often used concept to describe second order properties in 2D or 3D. In 2D it is given as:

$$H(L) = \begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial xy} \\ \\ \frac{\partial^2 L}{\partial xy} & \frac{\partial^2 L}{\partial y^2} \end{bmatrix}, \tag{1}$$

with $L(x, y)$ the image. In a pixel, the Hessian matrix can be used to calculate the second order derivative in any direction. In the spots on the contour of the stent, the second order derivatives are very high compared to the rest of the image. Therefore, we consider the following measures:

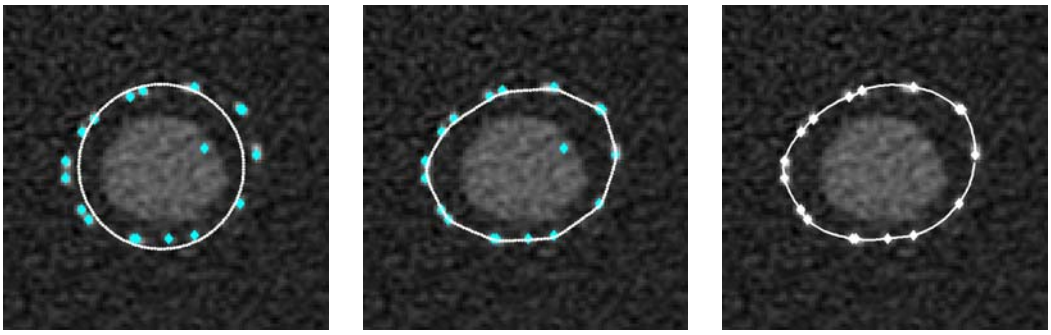
- The sum of the Eigenvalues. Also known as the Laplacian, or the trace of the Hessian matrix: $\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$.
- The product of the Eigenvalues. Also know as the Gaussian curvature or $\frac{\partial^2 L}{\partial x^2} \cdot \frac{\partial^2 L}{\partial y^2}$.

The functions $\frac{\partial^2 L}{\partial x^2}$ and $\frac{\partial^2 L}{\partial y^2}$ are calculated by convolving the image with (second order) Gaussian derivative kernels with $\sigma = 1.6$. For a pixel to be classified as a candidate point it has to be a local maximum and the Laplacian or Gaussian curvature in that point has to be larger than a certain threshold. This threshold is the main parameter of the methods and are empirically determined at 95 and 2000 for the Laplacian and Gaussian curvature respectively, where the pixelvalues are scaled at one unit per CT-value.

2.5 Clustering: Fitting

The first proposed method to segment the stent is to iteratively fit circles and ellipses to the stent, assuming that its shape does not deviate too much from an ellipse. The method starts by running the algorithm of 2.1.4 to produce an initial circle estimate. Next, the algorithm calculates which points are closer than $2 \cdot t_{\text{tol}}$ to this circle (where t_{tol} is the tolerance measure) and selects these points. A new circle is fit to the selected points and the process repeats itself until the circle does not change. Then the method will fit ellipses rather than circles and points must now be closer than t_{tol} to the ellipse. When the ellipse estimate is converged the selected points are the contour. See figure 6 for an example. The tolerance measure t_{tol} is the single parameter of this method; we used a value of $t_{\text{tol}} = 0.1mm$.

Fitting an ellipse and calculating the distance of a point to an ellipse are much more complex then for a circle. The algorithm we use to fit ellipses is proposed by Halir et al.[7] in 1998, and an improvement on the original algorithm of Fitzgibbon et al.[8] To calculate the distance of a point to an ellipse we applied the algorithm proposed by Eberly in 2004.[9]



(a) initial snake (b) converged snake (c) circular spline

Figure 7 – Illustration of the snake method applied to an example.

2.6 Clustering: Snake

Active contours (or snakes) are often used in medical imaging for segmentation. A general challenge of active contours is to influence the snake where it is in a homogeneous region. We have chosen to solve this problem by implementing the method of Xu and Prince: gradient vector flow[10] (GVF). Another solution is to use a balloon force such as in Cohen et al.[11] and more recently in Dedbleds-Mansard et al.[12] We have seen, however, that in our setting the balloon force can force the snake over the points, which allows the snake to grow unlimited until iteration stops.

The GVF method iteratively generates a vector field (the GVF field) to guide the snake. This field enables the snake to move into boundary concavities, gives the GVF snake a large capture range, and makes it insensitive to initialization. Especially the latter two features are expected to make this type of snake very robust and work well in our setting.

We generate a black image with white pixels on the locations of the detected points, and create a GVF field from this image. We initialize the snake by sampling points on the circle estimated by the algorithm of 2.1.4. This makes the method invariant to spurious points inside the contour, see figure 7 for an example. After the snake has converged, all points closer than 1.4 pixels to the snake are selected as the contour.

One drawback of snakes in general is their large amount of parameters. In addition to *elasticity*, *rigidity*, *viscosity*, *external force weight* we can also change the sample distance of the snake. On top of that, the resolution (number of pixels per mm) of the created image also has influence on its behaviour. In our experiments we used the slice's resolution because the resolution among different datasets only differs by a few percent. We used a relatively large rigidity to prevent the snake from leaking through the larger gaps between points. The appropriate rigidity value was found to be 20.

2.7 Clustering: Chopstick

We propose an algorithm that uses the internal geometry of the detected points to select the points on the contour of the stent. It is designed for this clustering problem in specific and has the advantage of an (in our opinion) intuitive working.

The first step consists of selecting a coarse set on the contour, which we will explain using a virtual stick. The stick is attached to a point found by the algorithm of 2.1.4 and aimed at the centre. We rotate the stick around the initial point until it touches another point, which we then take into our selection. We attach the stick to the new point and the process is repeated until we reach a point that we already selected. In figure 8 an example result is illustrated.

The next step consists of adding and removing points. Consider two points n_1 and n_2 that are connected by the stick in the first step. The algorithm searches for a point in between n_1 and n_2 that is suitable for selection. If such a point is present, it is taken into the selection; the stick is chopped in two pieces to connect n_1 and n_2 to the new point. After searching in between all points, we examine each point using its two direct neighbours

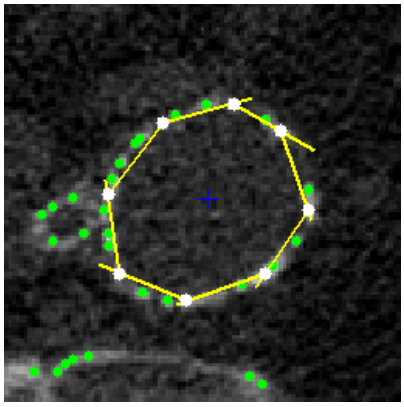
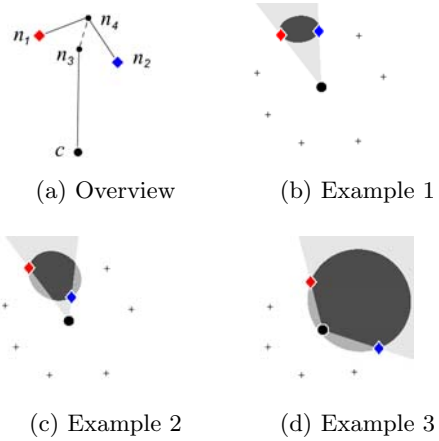


Figure 8 – Example result after finding a coarse contour.



(a) Overview (b) Example 1

(c) Example 2 (d) Example 3

Figure 9 – Illustration of the region in which points are found suitable by the chopstick method. The light, medium and dark shades of grey correspond to the first, second and combined criteria respectively.

(n_1 and n_2), and remove the point if it is not acceptable, using the same method. This process of adding and removing points is repeated until the selected set has converged.

The method to test the suitability of points is based solely on (the geometry of) the two reference points (n_1 and n_2) and the centre point. It adopts two simple criteria which are illustrated by three examples in figure 9. Consider c the centre point, p the candidate point, and $\overrightarrow{n_1\,n_2}$ and $\overline{n_1\,n_2}$ the vector and distance from n_1 to n_2 respectively. We calculate a point n_3 which lies exactly in between n_1 and n_2, and a point n_4 which lies on the normal of $\overrightarrow{n_1\,n_2}$ (see figure 8a).

The first criterion requires that the point must be in between the lines $\overrightarrow{c\,n_1}$ and $\overrightarrow{c\,n_2}$. This intuitively focusses at points "in line of sight". In the examples of figure 8 this is illustrated by the light grey area.

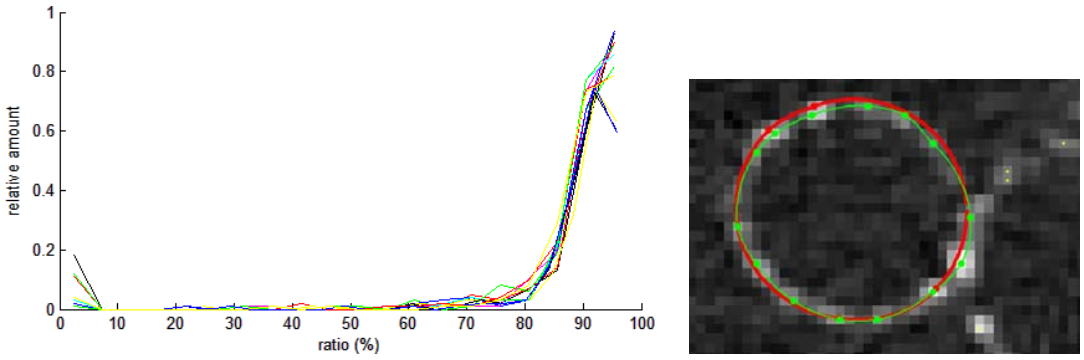
The second criterion requires the point to be inside an ellipse spanned between n_1 and n_2. An ellipse in general can be defined by two points and a constant D_{ellipse}. A point p is inside the ellipse if the distance from one ellipse-point to p and back to the other ellipse-point is smaller then D_{ellipse}. We position the two ellipse-points on the lines $\overrightarrow{n_1\,n_4}$ and $\overrightarrow{n_2\,n_4}$ respectively, and chose D_{ellipse} such that the ellipse just reaches n_1 and n_2. The distance from n_4 is equal for both ellipse-points and determines the eccentricity of the ellipse. We use a measure f to describe this distance, where $f = 0$ means both ellipse-points are in n_4 (the ellipse is a circle) and $f = 1$ means the points are in n_1 and n_2 respectively (the ellipse is maximally elongated). We want the shape of the ellipse to depend on $\overline{c\,n_3}$, being close to a circle when c is close, and elongated when c is far. We propose for f:

$$f = 1 - e^{-\overline{c\,n_3}\,/\,\overline{n_1\,n_2}} \tag{2}$$

The displacement of n_4 relative to n_3 is applied because points are more likely to be found on the outer side of the line $\overrightarrow{n_1\,n_2}$ due the circular nature of the contour. The distance of n_4 to n_3 is varied by the factor f of equation (2):

$$\overline{n_3\,n_4} = \frac{1}{4} \cdot \frac{\overline{c\,n_3}}{f} \tag{3}$$

We will shortly discuss the examples in figure 9 for clarification. Example 1 shows eight points on the contour and the region between n_1 and n_2 where points are justified according to the described criteria. In example two n_2's location is different; this point will be removed in a later stage of the algorithm. Example three illustrates a large gap between n_1 and n_2. It can be seen how the displacement of n_4 relative to n_3 moves the centre of the ellipse outward allowing points to be included that would not be found suitable if n_3 was the centre of the ellipse.



(a) (b)

Figure 10 – ***a:*** *Histogram illustrating the distribution of the ratios.* ***b:*** *An example result. The thick red line is the annotation. The green thin line the result for Gaussian curvature with chopstick.*

3. EXPERIMENTS

3.1 Setup

We have performed experiments to compare the methods described in the previous section. For this purpose, we randomly selected 200 slices from a set of scans of 23 patients. Each slice is centred at the stent, and tilted so that the slice is approximately perpendicular to the stent's centreline. In cases where two "legs" of the stent were visible, we randomly selected either the left or the right. All slices were then stored in a database file.

We build a MATLAB application to annotate the slices by hand. With this tool, we were able to load a database of slices, and select points on each slice through which a circular spine (see 2.1.3) was drawn to show the contour of the stent. An experienced radiologist annotated the 200 slices. We used the annotation data as our "gold standard" to compare the performance of the methods. Additionally, we created a training set of 50 slices which we used to determine the best parameter values of the methods. All the values of parameters and thresholds mentioned in the previous section have been determined using the trainingset.

The algorithms were applied to the slices in the database and compared to the annotation data. Both the annotated data and the algorithms produced a set of points on the contour of the stent. A circular spline was drawn through the points to make the contours continuous, which enabled us to calculate the area of the segmented stent, as well as compare the areas of the calculated contour and the annotated contour. We used the following performance measure:

$$r = \frac{A \cap M}{A \cup M}, \tag{4}$$

where A is the annotated area and M is the area that resulted from the selected method. We have calculated the ratio r of each annotated slice in the database and so produced an array of ratios.

3.2 Results and Discussion

In figure 10a the histograms of all 12 methods are shown to illustrate the distribution of r, the performance. From the overall shape it can be seen that the performance is mainly above 85%. In figure 10b an example result is shown against the annotation for a case with $r = 93\%$. It is clear that the result is good, which indicates that the (7%) error is due to inaccuracy of the annotation.

The error which we want to measure is not the accuracy of the points—the algorithm would always be more accurate than an annotator—but the *selection* of points. Because such errors are relatively rare, we chose not to simply average the array of ratios to compare the methods. Instead, we take the worst 30% and average it. In this way the difference in performance between the methods is much more clear.

In figure 11 the resulting performances are shown in a bar plot. We will first discuss the results of the point detection methods and then the results for the clustering methods.

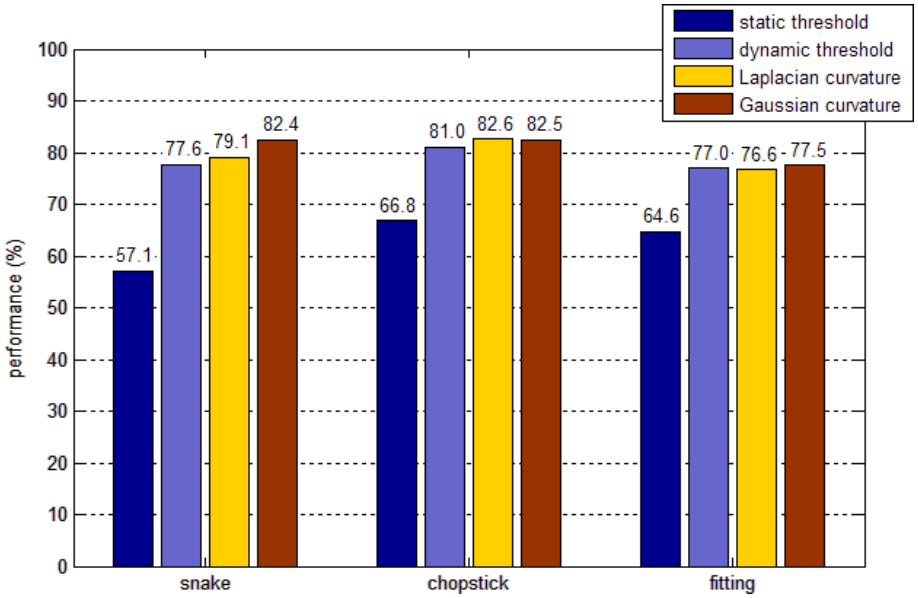


Figure 11 – Bar plot illustrating the performance of all combinations of methods

3.2.1 Point Detection Methods

It is clear that the performance of static thresholding is the worst. Although dynamic thresholding significantly improves the results, both Hessian based methods perform even better. It appears that the pointy nature of the structures on the contour of the stent is a very characteristic feature. We have also seen in experiments that the amount of spurious points on bone tissue or calcifications is significantly reduced for the Laplacian and Gaussian curvature compared to the two thresholding methods. We suspect that the characteristic property of the criterion is the cause of this.

Initially, we also considered the first Eigenvalue of the Hessian matrix as a measure, as it is the second order derivative in the direction where it it is largest. However, experiments revealed at an early stage that the other two Hessian based methods outperform this measure. It appears the Gaussian curvature detection method performs best. Although the Laplacian scores better in combination with the chopstick clustering method, the difference is so small that the significance is probably neglectable. It does suggest, however, that in combination with chopstick, Laplacian point detection works as well as Gaussian curvature point detection. It is interesting to see that the difference between the two Hessian based methods is so pronounced when combined with snake clustering.

3.2.2 Clustering Methods

It is clear that the fitting method performs poorly compared to the other methods. This suggests that the assumption that the stent's contour is shaped like an ellipse is insufficient. There are, however, specific situations (as in the example of figure 6) where fitting succeeds and the other clustering methods fail.

For each point detection method, the snake's performance is lower than that of chopstick. Although the difference is neglectable in combination with Gaussian curvature point detection, the difference is larger in combination with the other point detection methods. In combination with the static threshold the snake method is even outperformed by the fitting method. We suspect that although the snake *can* perform good, it is sensitive for spurious points and missing points. It could be that the performance can improve if more care is taken to tune all parameters, but as this is very time-consuming, we have not been able to do that.

We propose the combination of Gaussian curvature and the chopstick method, but remark that Laplacian point detection with chopstick clustering and Gaussian curvature point detection with snake clustering perform equally well.

4. CONCLUSIONS AND FUTURE WORK

We have proposed to segment stents in ECG-gated CTA data by detecting the stent in slices perpendicular to its centreline, by first detecting points and then select the points belonging to the stent by means of a clustering method.

We have discussed four point detection methods and three clustering methods and evaluated the performance of all combinations with an experiment. We have proposed a simple yet effective method for streak artifact invariance which is applied in the four point detection methods. Contrast fluids sometimes cause spurious points, but the clustering algorithms are all to some extend robust for such points. The proposed method is to detect points using the Gaussian curvature of the image, and to cluster the points using the proposed chopstick algorithm.

In future research we plan to use the proposed method and extend our approach to segment the stent in 3D by following the centreline of the stent. The set of segmented contour points can be tracked in the volumes that represent the different phases of the cardiac cycle and will enables us to measure the surface and diameter change during the cardiac cycle.

Additionally, we plan to research the possibilities in segmenting the stent directly from the 3D data. In recent years, vessels have successfully been segmented[13] using 3D differential geometry; the wires of a stent's metallic framework have similar geometrical structure to a narrow vessel. Therefore we expect that it is possible to segment the stent using a similar method.

REFERENCES

1. M. Prinssen, E. Buskens, and J. Blankensteijn, "The Dutch randomised endovascular aneurysm management (dream) trial. background, design and methods," *J. Cardiovasc Surg. (Torino)* **43**(3), pp. 379–84, 2002.
2. L. Flórez-Valencia, J. Montagnat, and M. Orkisz, "3d graphical models for vascular-stent pose simulation," *International Journal of Machine Graphics & Vision* **13**(3), pp. 235–248, 2004.
3. J. Lee, P. Beighley, E. Ritman, and N. Smith, "Automatic segmentation of 3d micro-ct coronary vascular images," *Medical Image Analysis* **11**, pp. 630–647, 2007.
4. D. Umbach and K. N. Jones, "A few methods for fitting circles to data," *IEEE Transactions on instrumentation and measurement.* **52**(6), pp. 881–1885, 2000.
5. M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine* **16**(6), pp. 22–38, 1999.
6. J. J. Koenderink, "The structure of images," *Biological Cybernetics* **50**(5), pp. 363–370, 1984.
7. R. Halír and J. Flusser, "Numerical stable direct least squares fitting of ellipses," in *Internatianal Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, 1998.
8. A. W. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least squares fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(5), pp. 476–480, 1999.
9. D. Eberly, "Distance from a point to an ellipse in 2D," 2004.
10. C. Xu and J. L. Prince, "Snakes, shapes and gradient vector flow," *IEEE Transactions on Image Processing* **7**(3), pp. 359–369, 1998.
11. L. D. Cohen, "On active contour models and balloons," *Computer Vision, Graphics, and Image Processing: Image Understanding* **53**(2), pp. 211–218, 1991.
12. C. Desbleds-Mansard, A. Anwander, L. Chaabane, M. Orkisz, B. Neyran, P. C. Douek, and I. E. Magnin, "Dynamic active contour model for size independent blood vessel lumen segmentation and quantification in high-resolution magnetic resonance images," in *Computer Analysis of Images and Patterns: 9th International Conference, CAIP 2001 Warsaw, Poland, September 5-7, 2001, Proceedings*, 2001.
13. A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," *Lecture Notes in Computer Science* **1496**, pp. 130–137, 1998.