# FAST TEMPORAL TRACKING AND 3D RECONSTRUCTION OF A SINGLE CORONARY VESSEL

*Ifeoma Nwogu, Liana Lorigo*

State University of New York at Buffalo,
Computer Science and Engineering Department

## ABSTRACT

Vessel extraction and vessel motion estimation from X-ray angiograms has been a challenging computer vision problem for several years. We have developed a fast and accurate method for extracting and tracking intravascular imaging data from X-ray angiograms. We accomplish this by reconstructing a moving 3D vessel, which contains more information than the static 2D snapshot image. Our approach involves identifying the vessel-of-interest in two biplane images, abstracting them into centerlines, and tracking them in ensuing images using deformable templates and graph techniques for optimization. When tested on fifteen patient datasets, the computational time was approximately 5 seconds per vessel per frame for vessels of length 80-100mm.

***Index Terms***— X-ray angiocardiography, Motion analysis, Blood vessels

## 1. BACKGROUND AND PURPOSE

With the wide availability and utility of X-ray coronary angiography, it is still very important to continue to investigate methods to improve coronary disease analysis using X-rays. X-ray based quantitative coronary analysis (QCA) can be optimized by incorporating data from additional images in the angiogram sequence. For this reason, we propose a fast temporal vessel tracking algorithm to provide additional information on the same vessel over time.

This motion estimation procedure primarily focuses on tracking the centerlines of the blood vessels. We briefly review previous two-dimensional (2-D) and three-dimensional (3-D) tracking processes. In previous 2-D cardiac motion estimation techniques, optical flow methods [1], Kalman Snakes [2] and deformable lines [3] were used to track the arterial centerline. In the 3-D techniques, a 3-D model of the coronary artery centerlines is constructed at one time frame and then deformed to match up with the angiogram image pairs at later frames. Shechter et al[4], minimized B-spline curve energies, while Chen and Carrol [5] found temporal correspondences by minimizing an arterial shape function. Both the 2-D and 3-D methods have yielded promising results over the years, but are still not clinically practical due to intensive computational requirements, especially the 3-D methods [6]. Also, regarding the 2-D methods mentioned above, tracking becomes quite complex in regions of multiple vessel overlap.

We present an efficient way to abstract a vessel into its centerline and an optimal way to track the vessel over time. We also present a brief overview of our vessel sizing and 3D reconstruction processes, but these are not currently the main thrust of the paper.

## 2. METHOD

The end-to-end process can be divided into three major stages: (1) The prerequisite stage where a vessel is identified and abstracted into its centerline; (2) the tracking stage where the centerline is approximated into a polyline and tracked using deformable templates and (3) the post-tracking stage where the 3D vessel is reconstructed.

### 2.1. Tracking prerequisites

The process is initiated with the user providing the proximal and distal points on the vessel of interest in the starting frame of the X-ray angiography sequence. A region-of-interest (ROI) around the vessel is then established.

#### *2.1.1. Vessel Enhancement.*

Because of the relatively low Signal-to-Noise Ratio (SNR) the vessels need to be enhanced,while suppressing other signals due to bone, tissue and noise. We used an enhancement filter modified from [7], based on the principle that the second derivatives of an image give strong responses on blobs or ridges. A vessel can be viewed as a ridge with its local extremum occurring at the centerline. The vessel ROI, $I$, is convolved with a Gaussian kernel of standard deviation $\sigma$ centered at pixel $p(x, y)$. The Hessian matrix of the convolved sub-image is thus given as:

$$\mathbf{H}(I, \sigma, p) = \begin{pmatrix} \frac{\partial^2 I_N(\sigma,p)}{\partial x^2} & \frac{\partial^2 I_N(\sigma,p)}{\partial y \partial x} \\ \frac{\partial^2 I_N(\sigma,p)}{\partial x \partial y} & \frac{\partial^2 I_N(\sigma,p)}{\partial y^2} \end{pmatrix} \quad (1)$$

$\lambda_1$ and $\lambda_2$ are the eigenvalues of the Hessian matrix $\mathbf{H}$ where

$\|\lambda_1\| << \|\lambda_2\|$ while $\mathbf{v_1}$ and $\mathbf{v_2}$ are the eigenvectors corresponding to $\lambda_1$ and $\lambda_2$ . Unlike its previous implementations, we use only the eigenvectors of the Hessian matrix for vessel enhancement.

At each pixel $p$, the potential edge of the vessel is computed by adding the vector $\pm\sigma\mathbf{v_1}$, where $\mathbf{v_1}$ represents the direction perpendicular to the ridge direction. The vessel edge strength in either direction $\nabla I(\sigma, p + \pm\sigma)$ is then computed using the standard gradient methods. The edge strength thus provides a measure of the likelihood of $p$ being a ridge pixel and $\sigma$ being the true vessel width. The filter response is therefore defined as the smaller angle between $\mathbf{v_1}$ and edge strength $\nabla I(.)$ computed to the left and the right of $p$.

*2.1.2. Centerline Extraction.*

We used the traditional non-maximal suppression and hysteresis thresholding algorithms introduced into the computer vision community as the post-processing routines in Canny's Edge Detector [8], to better define the centerlines.

Lastly, we used standard computer vision edge linking techniques to connect the remaining vessel "stubs" to complete the centerline extraction process An example of the extraction process on a vessel ROI is shown in figure 1.

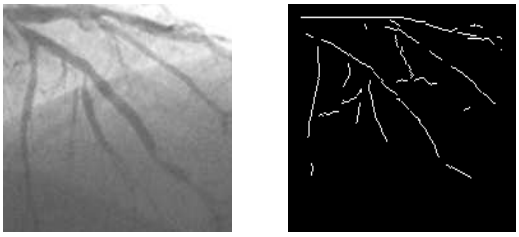

**Fig. 1**. The left image shows an original ROI image with the vessel of interest in the middle of the image and the right image shows the resulting centerlines.

*2.1.3. Vessel Size Estimation.*

Once the centerline has been successfully extracted, edge candidates are computed using profiles, perpendicular to the centerline. Each profile is de-noised and the first derivatives of the profile are obtained using central differences. The gradient points become candidate edge points.

Because the shape of a vessel edges closely follow the orientation of its centerline, the centerline points are deformed to the most appropriate edge candidates to form a left (and right) contour line, as a smoothness constraint is imposed on the deformation. The vessel size estimate is obtained as the distance between the left and right edge points.

## 2.2. Vessel Tracking

The first step in performing the actual motion tracking is to approximate the centerline into a polyline which involves fitting straight line segments (with a pre-specified tolerance) to the centerline points. The process finds the size and position of the maximum deviation of the line that joins the endpoints of the curve. If the maximum deviation exceeds the allowable tolerance, the centerline is broken at the point of maximum deviation and the test is repeated. In this manner the entire centerline is broken down to line segments, which adhere to the original data with the pre-specified tolerance.

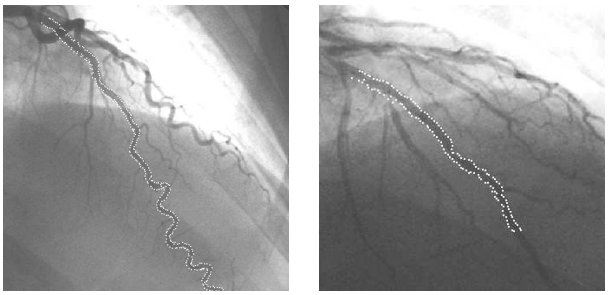

**Fig. 2**. Extracted contours from a vessel in region with multiple vessel overlap (see top) and bifurcations; from a vessel with narrowing due to stenosis.

*2.2.1. Fast Polyline Tracking.*

So far, the vessel features have been obtained in one frame. The vertices of the polyline are selected as anchor points and they are the key features to be tracked in the algorithm.

Having a priori information about the extent of motion of the coronary vessels, we then select a larger ROI in the ensuing image (to accommodate vessel motion) and in a similar fashion as before, we reduce the image to its centerline abstraction using only the value of $\sigma$ estimated from the previous image as the average vessel width. This significantly reduces the computational intensity, because we only perform a single scale search, unlike in the first image.

A square search neighborhood is then placed around each vertex of the polyline as in [3]. We select a relatively large search space, a neighborhood window $W_N$ of 61 x 61 pixels where the original images were 512 x 512. All centerline points in the search space are candidate vertices of the new polyline.

The candidate vertices represent the end points of candidate segments. These segments are treated as the nodes of a graph while the angles between adjacent segments correspond to the edges between the nodes of the graph.

The goal of this section of the algorithm is to perform a nonrigid deformation on the original polyline $\varphi$ into a new polyline $\varphi'$. We perform the deformation by first using a spring model, which introduces a penalty for stretching the

line segment from one frame to the next.

$$e_1(\varphi, \varphi') = \frac{(\|\varphi'\| - \|\varphi\|)^2}{\|\varphi'\| + \|\varphi\|} \quad (2)$$

Next, the shape of the polyline is constrained by introducing a penalty for changing the angle $\theta$ between 2 adjacent segments of the polyline, so that:

$$e_2(\theta, \theta') = \frac{(\theta' - \theta)^2}{\theta' + \theta} \quad (3)$$

Hence, the total cost of deforming two adjacent line segment $\varphi_i$ and $\varphi_j$ to $\varphi_i'$ and $\varphi_j'$ is:

$$E = \alpha e_1(\varphi_i, \varphi_i') + \beta e_2(\theta, \theta') \quad (4)$$

Having assigned the cost value above to the edges of the graph, we use Dijkstra's shortest path technique to find the minimum cost of deformation. The nodes on the shortest path became the line segments of the new polyline in the next image frame.

The true centerline is then obtained by performing a quick search in the vicinity of the estimated polyline. A new polyline is created from the true centerline - this step is required because the estimated polyline could potentially fall outside the vessel if there are local deformations unaccounted for by the global polyline. The process is repeated for every image in the angiography sequence until at least one of the search spaces is empty in a new image.

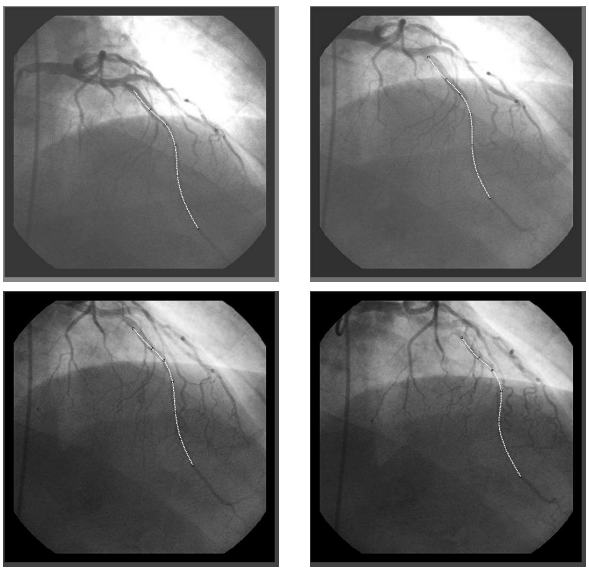

**Fig. 3**. The results of 2D+t motion tracking are shown. From top left to bottom right are results of tracking in 15 frame intervals, i.e at frame $t_0$, $t_15$, $t_30$ and $t_45$. Observe the position of coronary tree relative to the diaphragm.

### 2.3. Vessel 3-D Reconstruction

Once the vessel-of-interest has been tracked in the two biplane sequences, angulation information, the $R$ and $t$ matrices are retrieved from the image gantries. One of the 2 views is selected as the primary, and using the centerline points of the primary, epipolar lines are constructed in the secondary image. The intersections of the epipolar lines with the points on the vessel centerline are taken as the correspondence points. A 3-D centerline is reconstructed using standard epipolar geometry constraints. The vessel lumen is generated from the centerline using the previously calculated centerline radius values.

Iso-correction is performed by optimizing the geometry at which the 2-D re-projection of newly generated 3-D vessel best aligns with the original image representation in the two views, using the simplex algorithm for the optimization. More details on this process can be found in [9].

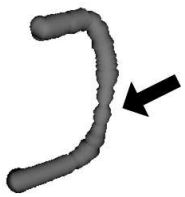

**Fig. 4**. A 3-D reconstruction of a vessel from one time frame, showing a potential narrowing. Similar views from more time frames can confirm/reject and better quantify such a hypothesis.

## 3. RESULTS

We tested our algorithm on fifteen patient datasets, ten from single plane angiograms and five end-to-end from biplane angiograms. The patients were imaged using a Toshiba DFP-2000A/AS for single plane and Toshiba DFP-2000A/3 for biplane. Both systems have FOV of $172\mu$.

The computational platform used in testing was a Pentium 4 running at 3.0GHz. All implementation was done in C++, with additional FLTK (Fast Light Toolkit) libraries for the front-end interface and VTK (The Visual Toolkit) libraries for visualization.

The test datasets involved a combination of vessel types with lengths ranging from about 80-110mm. On average, there were about 70 frames in a sequence where the vessel of interest was clearly visible and could be tracked. Some of the average processing times per vessel per frame are given in Table 1 below.

| Major Step | Average Processing Time |
|---|---|
| Vessel enhancement (single-scale) | $\approx$ 1s |
| Centerline extraction | $<$ 1s |
| Vessel sizing | $\approx$ 0.5-3s |
| Polyline tracking | $\approx$ 0.5-3s |
| Vessel reconstruction and optimization | $<$ 2s |

**Table 1**. Computational times per vessel per frame.

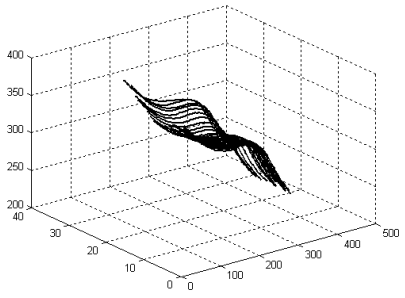


**Fig. 5**. Trajectory of a coronary vessel centerline over time for one complete cardiac phase. Dense dark region represents the slower moving diastole phase.

## 4. DISCUSSION

We have presented a close to real-time algorithm to characterize, track and reconstruct a 3-D moving vessel over time by developing new algorithms and enhancing already existing ones from the field of computer vision. We believe that the presented method can be readily used clinically as a precursor to optimizing many intravascular QCA processes such as vessel sizing for stenosis evaluation, plaque characterization etc., from X-ray angiograms.

As a next step, we intend to test our method against previous 2-D methods and also against the more complicated, but slower 3-D methods discussed in the paper, to compare the quality and speed of tracking. We also plan to validate the algorithm quantitatively using phantoms with known parameters.

One major practical advantage that our proposed algorithm has over previous approaches is its strong potential for application in clinical rather than only research environments.

## 5. REFERENCES


[1] J. Meunier, M. Bourassa, M. Bertrand, M. Verreault, and G.E. Mailloux, "Regional epicardial dynamics computed from coronary cineangiograms," *Compt. Cardiol.*, pp. 307–310, 1989.

[2] R. Curwen, A. Amini, J. Duncan, and F. Lee, "Tracking vascular motion in x-ray image sequences with kalman snakes.," *Compt. Cardiol.*, pp. 109–112, 1994.

[3] M.P. Dubuisson-Jolly, C.C. Liang, and A. Gupta, "Optimal polyline tracking for artery motion compensation in coronary angiography," in *ICCV98*, 1998, pp. 414–419.

[4] G. Shechter, F. Devernay, E. Coste-Maniere, A. Quyyumi, and E. R. McVeigh, "Three-dimensional motion tracking of coronary arteries in biplane cineangiograms," *IEEE Trans. on Med. Imag.*, vol. 22, no. 4, pp. 493–503, 2003.

[5] S. Y. Chen and J. D. Carroll, "Kinematic and deformation analysis of 4-d coronary arterial trees reconstructed from cine angiograms," *IEEE Trans. on Med. Imag.*, vol. 22, no. 6, pp. 710–721, 2003.

[6] C. Blondel, G. Malandain, R. Vaillant, and N. Ayache, "Reconstruction of coronary arteries from one rotational x-ray projection sequence," Tech. Rep., INRIA, May 2004.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," *Medical Image Computing and Computer-Assisted Intervention - Miccai'98*, vol. 1496, pp. 130–137, 1998.

[8] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[9] K.R. Hoffmann, A.M. Walczak, and P.B. Noel, "3d reconstruction of the carotid artery from two views using a single centerline," pp. 177–182.