City Research Online

City St George's, University of London

Citation: Slabaugh, G. G., Kong, K., Unal, G. B. & Fang, T. (2007). Variational Guidewire Tracking Using Phase Congruency. Paper presented at the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2007, 29-10-2007 - 02-11-2007, Brisbane, Australia. doi: 10.1007/978-3-540-75759-7_74

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/4413/

Link to published version: https://doi.org/10.1007/978-3-540-75759-7_74

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

Variational Guidewire Tracking Using Phase Congruency

Greg Slabaugh[1], Koon Kong[2], Gozde Unal[1], Tong Fang[1]

[1] Intelligent Vision and Reasoning Department, Siemens Corporate Research, USA
{greg.slabaugh|gozde.unal|tong.fang}@siemens.com
[2] School of Electrical and Comp. Engineering, Georgia Institute of Technology, USA
kykong@gatech.edu

Abstract. We present a novel method to track a guidewire in cardiac x-ray video. Using variational calculus, we derive differential equations that deform a spline, subject to intrinsic and extrinsic forces, so that it matches the image data, remains smooth, and preserves an a priori length. We analytically derive these equations from first principles, and show how they include tangential terms, which we include in our model. To address the poor contrast often observed in x-ray video, we propose using phase congruency as an image-based feature. Experimental results demonstrate the success of the method in tracking guidewires in low contrast x-ray video.

1 Introduction

Endovascular interventions are becoming increasingly more common in the treatment of arterial disease like atherosclerosis. In such procedures, a guidewire is placed in groin and advanced towards the heart. Critical to this process is accurate placement of the guidewire with respect to the vascular anatomy, which is typically imaged using x-ray fluoroscopy. However, placement is often difficult due to complexity of the vasculature, patient motion, and the low signal to noise ratio of the video that results from trying to minimize the radiation exposure to the patient. In this paper, we present a method to track a guidewire in cardiac x-ray video. Tracking the guidewire has many applications, including interventional navigation and adaptive image enhancement of the guidewire.

While there exist numerous papers on the subject of line detection in noisy images, there is relatively little literature devoted to the more specific topic of guidewire tracking, which is perhaps unexpected given the clinical importance of endovascular interventions. Palti-Wasserman et al. [1] were to our knowledge the first to consider the problem; in their approach the guidewire is modeled using a second degree polynomial extracted from consecutive frames of a video. Recent work by Baert et al. [2] models the guidewire using a spline, and then optimizes the spline position numerically using Powell's direction set method. The optimization is designed to deform the spline so that it has minimal length, remains smooth, and matches the guidewire position in the image.

Several authors have considered evolving a contour subject to various intrinsic and image-based forces; snakes [3] [4] being a classic example. When the contour is represented using a spline interpolated from discrete control points, the problem then becomes one of evolving the control points, which in turn evolve the spline. Typically, authors consider closed contours [5] or open contours with boundary conditions [6], such as forcing the endpoints to be fixed or have mirror symmetry. However, such boundary conditions are not suitable for guidewire tracking. When evolving closed contours, tangential forces on the contour are typically ignored as they do not change the contour geometry. However, these terms have an effect on an open contour and therefore should be addressed.

1.1 Our contribution

Our work is inspired by [1] and [2] but has significant differences. We also model the geometry of the guidewire as a spline, defined as a smooth curve that interpolates control points. We compose our energy functional of three terms, one designed to force the guidewire to match the edge-detected pixels in the image, another to keep the spline smooth, and a third designed to retain an a priori length (as opposed to the minimal length of [2]). Unlike [1] [2], we analytically derive, using variational calculus, differential equations that describe the flow of the spline to minimize the energy functional and achieve a locally optimal position of the spline on the current frame. By sampling the spline sufficiently, we obtain an over-determined system of linear equations which can be inverted to relate the motion of the spline to that of the control points. This then gives us a simple mechanism to evolve the control points of an open spline without enforcing any unnatural boundary conditions. We derive the differential equations from an energy formulation, and from this we see tangential terms that are typically ignored for closed contour evolutions. For the image-based terms used to align the spline with the guidewire, we propose the use of phase-congruency [7], which is able to accurately detect guidewire pixels in x-ray images with low contrast.

2 Variational formulation

In this section we derive the evolution of the spline using variational calculus. We represent the guidewire as an open curve $\mathbf{C} = [x(s), y(s)]$ in the image plane, where $s \in [0, L]$ is an arc length parameter and L is the contour length. We begin by defining the energy E of the curve as

$$\begin{aligned} E(\mathbf{C}) &= w_1 \cdot \text{data} + w_2 \cdot \text{smoothness} + w_3 \cdot \text{length constraint} \\ &= w_1 \int_{\mathbf{C}} F ds + w_2 \int_{\mathbf{C}} ds + w_3 (\int_{\mathbf{C}} ds - L_0)^2, \end{aligned} \tag{1}$$

where w_1, w_2, and w_3 are constants used to weigh the terms relative to one another. The data term will require that the spline adhere to the features detected from the x-ray image and is based on $F(x, y)$, which is a conformal factor computed from a feature map of the x-ray image. The smoothness term will require

the curve to be smooth, and the length constraint term penalizes the curve's length from deviating from an a priori length L_0.

2.1 Regularization

Let us consider the second term of Equation 1 first. Taking the partial derivative with respect to an independent time parameter t, and reparameterizing with $p \in [0,1]$, gives

$$\begin{aligned}\frac{\partial}{\partial t} w_2 \int_{\mathbf{C}} ds &= w_2 \frac{\partial}{\partial t} \int_0^1 ||\mathbf{C}_p|| dp = w_2 \int_0^1 \frac{\partial}{\partial t} < \mathbf{C}_p, \mathbf{C}_p >^{\frac{1}{2}} dp \\ &= w_2 \int_0^1 \frac{< \mathbf{C}_{pt}, \mathbf{C}_p >}{||\mathbf{C}_p||} dp = w_2 \int_0^1 < \mathbf{C}_{pt}, \mathbf{T} > dp \\ &= w_2 < \mathbf{C}_t, \mathbf{T} > |_{p=0}^1 - w_2 \int_0^L < \mathbf{C}_t, \kappa \mathbf{N} > ds, \end{aligned} \tag{2}$$

where κ is the curvature of the curve, $\mathbf{T}$ is the tangent, $\mathbf{N}$ is the normal, and $<,>$ denotes an inner product. Thus, to minimize the length of the curve, we evolve the curve using

$$\frac{\partial \mathbf{C}}{\partial t} = w_2 \kappa \mathbf{N} + w_2 \delta(p) \mathbf{T} - w_2 \delta(p-1) \mathbf{T} \tag{3}$$

where δ is a delta function. Intuitively, this flow indicates that to minimize the curve's length, one can simply move each point in the normal direction weighted by curvature, and at the endpoints of the curve, move the curve inwards along the tangent. Note that these tangential terms cancel in closed contour evolutions, for which $p = p - 1$. Evolving the contour with this differential equation will cause it to both shrink and become smoother. Later, our length preserving term will force the contour to retain an a priori length, so this term will effectively regularize the contour, keeping it smooth.

2.2 Geodesic flow

Now let us consider the first term of Equation 1. Taking the partial derivative with respect to an independent time parameter t, gives

$$\begin{aligned}\frac{\partial}{\partial t} w_1 \int_{\mathbf{C}} F ds &= w_1 \int_0^1 \frac{\partial}{\partial t} F ||\mathbf{C}_p|| dp = w_1 \int_0^1 \left(F \frac{\partial}{\partial t} < \mathbf{C}_p, \mathbf{C}_p >^{\frac{1}{2}} + \frac{\partial F}{\partial t} ||\mathbf{C}_p|| \right) dp \\ &= w_1 \int_0^1 F < \mathbf{C}_{pt}, \mathbf{T} > dp + w_1 \int_0^L < \nabla F, \mathbf{C}_t > ds \\ &= w_1 F < \mathbf{C}_t, \mathbf{T} > |_{p=0}^1 - w_1 \int_0^L F < \mathbf{C}_t, \kappa \mathbf{N} > ds + w_1 \int_0^L < \nabla F, \mathbf{C}_t > ds \end{aligned} \tag{4}$$

Thus, to minimize the conformally weighted length of the curve, we get

$$\frac{\partial \mathbf{C}}{\partial t} = w_1 F \kappa \mathbf{N} - w_1 \nabla F + w_1 \delta(p) F \mathbf{T} - w_1 \delta(p-1) F \mathbf{T} \tag{5}$$

This derivation is similar to that of [4] except we now consider the tangential terms that act on the open contour.

2.3 Length preserving flow

Finally, let us consider the last term of Equation 1. Taking the partial derivative with respect to an independent time parameter t, gives

$$\frac{\partial}{\partial t} w_3 (\int_{\mathbf{C}} ds - L_0)^2 = 2w_3 (\int_{\mathbf{C}} ds - L_0) \frac{\partial}{\partial t} \int_{\mathbf{C}} ds \tag{6}$$

Using the result from the first term derivation in Section 2.1, we get the curve evolution,

$$\frac{\partial \mathbf{C}}{\partial t} = 2w_3 (\int_{\mathbf{C}} ds - L_0) \left[\kappa \mathbf{N} + \delta(p)\mathbf{T} - \delta(p-1)\mathbf{T}\right] \tag{7}$$

2.4 Complete flow

Combining Equations 3, 5, and 7, we get the final curve evolution,

$$\begin{aligned} \frac{\partial \mathbf{C}}{\partial t} &= -w_1 \nabla F + \kappa (w_2 + w_1 F + 2w_3 (\int_{\mathbf{C}} ds - L_0)) \mathbf{N} \\ &+ \left[\delta(p)(w_2 + w_1 F + 2w_3 (\int_{\mathbf{C}} ds - L_0)) + \delta(p-1)(-w_2 - w_1 F - 2w_3 (\int_{\mathbf{C}} ds - L_0))\right] \mathbf{T} \end{aligned} \tag{8}$$

2.5 Spline representation

Equation 8 is the contour evolution equation and is independent of the representation of the contour. That is, the equation would apply to any open contour representation, whether it be described as a polyline, implicit curve, using Fourier descriptors, etc. We choose to model the curve as a spline, with control points $\mathbf{P} = [\mathbf{P}_1 \dots \mathbf{P}_N]^T$. Our objective is to derive an equation to evolve the control points, which then update the contour to track the guidewire. To do this, we must relate the differential motion of the control points to that of the contour. This depends on the spline representation.

We model the contour geometry using a uniform rational B-spline [8]. In this representation, the contour is represented by M segments that interpolate the $N = M + 3$ control points. In this paper, we set $M = 2$ and thus $N = 5$. This results in a curve with enough degrees of freedom to bend and follow the guidewire, but not too many to result in spurious local minima. The jth segment is a weighted combination of four control points, as $\mathbf{C}_j(p) = \sum_j^{j+3} B_j(p)\mathbf{P}_j$, where $j = 1 \cdots M$, $p \in [0, 1]$, and is a parametrization variable used to sample B_j, which are third order blending functions. We would like to this equation to express $\mathbf{P}_j$ as a function of $\mathbf{C}_j(p)$ and differentiate, to yield a differential relationship describing how the motion of the curve segment affects the control points. For example, consider the case when we have spline consisting of $M = 2$

segments, corresponding to $N = 5$ control points, where we sample each segment $L = 4$ times. This results in the system of equations:

$$\begin{bmatrix} \mathbf{C}_1(p_1) \\ \mathbf{C}_1(p_2) \\ \mathbf{C}_1(p_3) \\ \mathbf{C}_1(p_4) \\ \mathbf{C}_2(p_1) \\ \mathbf{C}_2(p_2) \\ \mathbf{C}_2(p_3) \\ \mathbf{C}_2(p_4) \end{bmatrix} = \begin{bmatrix} a(p_1) & b(p_1) & c(p_1) & d(p_1) & 0 \\ a(p_2) & b(p_2) & c(p_2) & d(p_2) & 0 \\ a(p_3) & b(p_3) & c(p_3) & d(p_3) & 0 \\ a(p_4) & b(p_4) & c(p_4) & d(p_4) & 0 \\ 0 & a(p_1) & b(p_1) & c(p_1) & d(p_1) \\ 0 & a(p_2) & b(p_2) & c(p_2) & d(p_2) \\ 0 & a(p_3) & b(p_3) & c(p_3) & d(p_3) \\ 0 & a(p_4) & b(p_4) & c(p_4) & d(p_4) \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \\ \mathbf{P}_5 \end{bmatrix} \tag{9}$$

where $a(p) = -p^3 + 3p^2 - 3p + 1$, $b(p) = 3p^3 - 6p^2 + 4$, $c(p) = -3p^3 + 3p^2 + 3p + 1$, and $d(p) = p^3$ are the elements in the blending function.

This system of equations takes the form $\mathbf{C} = B\mathbf{P}$, where $\mathbf{C}$ is a ML x 2 matrix, B is a ML x N matrix, and $\mathbf{P}$ is a N x 2 matrix. As long as the total number of samples $ML \geq N$, the system is (possibly over-) determined and we can express $\mathbf{P}$ as a function of $\mathbf{C}$ using the pseudo-inverse, $\mathbf{P} = (B^T B)^{-1} B^T \mathbf{C}$.

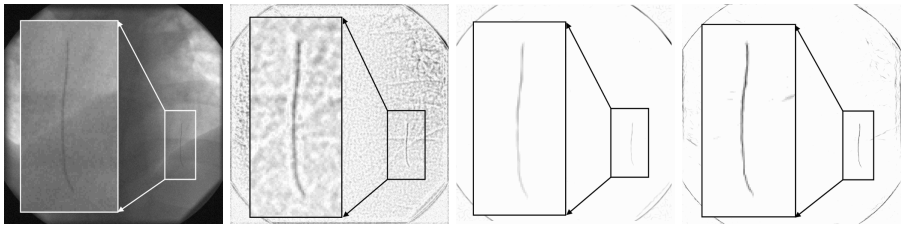
Thus, we can write the evolution of the control points simply as $\frac{\partial \mathbf{P}}{\partial t} = (B^T B)^{-1} B^T \frac{\partial \mathbf{C}}{\partial t}$, or

$$\begin{aligned} \frac{\partial \mathbf{P}}{\partial t} = (B^T B)^{-1} B^T \Bigg\{ & -w_1 \nabla F + \kappa (w_2 + w_1 F + 2w_3 (\int_{\mathbf{C}} ds - L_0)) \mathbf{N} \\ & + \left[\delta(p)(w_2 + w_1 F + 2w_3 (\int_{\mathbf{C}} ds - L_0)) + \delta(p-1)(-w_2 - w_1 F - 2w_3 (\int_{\mathbf{C}} ds - L_0)) \right] \mathbf{T} \Bigg\}. \end{aligned} \tag{10}$$

We note that the matrix $B^T B$ has size N x N, which is typically quite small, so the pseudo-inverse can be computed efficiently. Equation 10 gives us a differential equation that evolves the control points $\mathbf{P}$ in order to reposition the spline $\mathbf{C}$ so that it fits the data, remains smooth, and retains an a priori length.

3 Phase congruency

We base the data term in our technique on a function F that is computed using phase-congruency for guidewire detection. As discussed in [7], phase congruency is a dimensionless measure of feature significance that is less sensitive to contrast than differential techniques (gradient, Hessian, etc.). At an edge in an image, phase information is locally congruent, and the degree of this congruency can serve as an edge detector response. In [7], phase congruency is computed over multiple scales and orientations via a wavelet technique using log Gabor functions. Through experimentation, we use three scales, 6 orientations, $\sigma = 0.7$ for the Gaussian in the log-Gabor function, and $k = 7.5$ standard deviations for the noise energy threshold (see [7] for an explanation). In Figure 1, we provide an comparison of the phase congruency detector vs. the edgemaps produced using the Mexican-hat operator of [1] and the coherence-enhancing diffusion followed



(a) (b) (c) (d)

Fig. 1. An example image (a) and its corresponding feature map computed using the technique of [1] (b), [2] (c), and phase congruency (d). Notice the stronger contrast of the guidewire in the phase congruency result.

by Hessian computations of [2]. We observe that the phase congruency result has a sharp edge response and is not plagued by clutter.

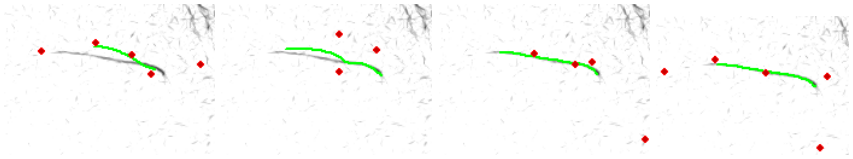
Let E be the edge response computed by phase congruency. We then form $F = \frac{1}{1+E^2}$. We compute ∇F using a smoothed derivative operator, and then run a GVF diffusion [9] on ∇F to increase the capture range of the gradient field.

4 Experimental results

We manually initialize the contour on the first frame of the sequence by clicking five control points, and the length of this contour is the a priori length L_0. We then execute Equation 10 using $w_1 = 1$, $w_2 = 1$, $w_3 = 0.1$, for a fixed number of iterations, 300 in our case. In our C++ implementation, 300 iterations typically takes approximately 175 milliseconds to run. An example is provided in Figure 2. Note that the spline updates its position to lock on to the guidewire detected using phase congruency, and also expand its length to match a priori length (105 pixels in this case) while remaining smooth. Upon convergence, we advance to the next frame, and use the cross-correlation technique of [2] to shift the spline to a new location, providing the initialization for the next frame. We do this because when the frame rate is low, the spline can jump from one location to the next on the image. The method then continues this process for the entire video.

We have tested our technique on 158 images coming from three different video sequences. We show the results of our tracking algorithm in Figure 3 for three consecutive frames from each sequence. The tracking is successful, matching the spline position with the guidewire in each image. The spline remains smooth but follows in the data, and the a priori length constraint discourages the spline from shrinking or expanding from known length (100, 105, 95 pixels, respectively, for these three examples).

Once initialized, the tracking is automatic. However, we have observed a small number of failures in the tracking, mainly due to motion blur and clutter.



(a) (b) (c) (d)

Fig. 2. Spline evolution (zoomed in to a region of interest). Given the initial spline (a), we evolve the control points, producing an intermediate result (b) and the final result upon convergence (c). We find it helpful to periodically recompute the control points from the contour, as shown in (d) using the spline from (c). We show the contour evolution overlaid on the phase congruency image in this figure. The spline is shown in green and the control points in red. Some control points have moved off the region of interest and not visualized in (b) and (c).

In the motion blur case, the guidewire is not distinctly visible, and the phase-congruency image does not provide sufficient information. In the clutter case, edges corresponding to anatomic structures or surgical instruments near the guidewire look like the guidewire, and the tracker starts to follow these false edges instead. Out of the 158 frames, the tracker failed for 10 frames, for an accuracy of 93%. These results are comparable to [2]; however, a detailed comparison would require running both algorithms on the same data, a subject left for future work. In the case of a tracking failure, the user may reinitialize the contour.

5 Conclusion

This paper presented a variational approach to guidewire tracking in cardiac x-ray video. We derived analytic equations to evolve the control points of a spline in order for the spline to match the image data, remain smooth, and preserve its length, and demonstrated the method's usefulness by tracking guidewires in several endovascular x-ray videos. While further experimentation is ongoing, our current experiments demonstrate much promise for this method to accurately track the guidewire, even in low-contrast videos. Further work on this method will include investigations into increasing robustness and detailed comparisons with alternate techniques, as well as modeling the spline as a 3D curve projected into multiple x-ray images. We believe the theory behind this work is quite general and applicable to numerous spline optimization problems.

References

1. Palti-Wasserman, D., Brukstein, A., Beyar, R.: Identifying and Tracking a Guide Wire in the Coronary Arteries During Angioplasty from X-Ray Images. IEEE. Trans. on Biomedical Engineering **44**(2) (1997) 152–164
2. Baert, S., Viergever, M., Niessen, W.: Guide-Wire Tracking During Endovascular Interventions. IEEE. Trans. on Medical Imaging **22**(8) (2003) 965–972

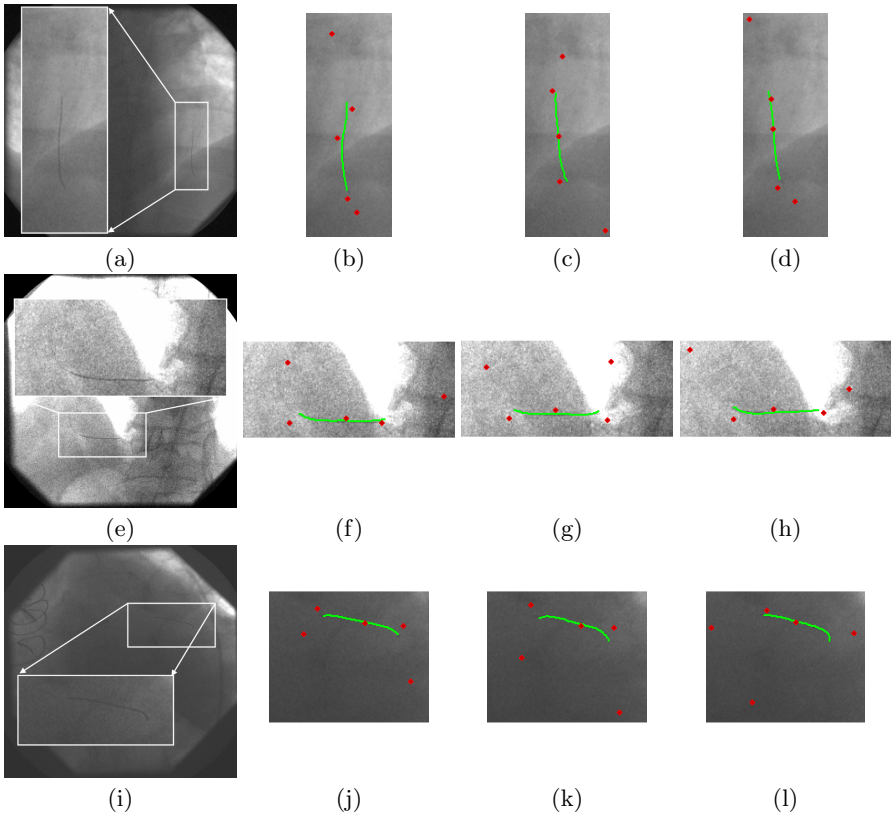


Fig. 3. Results of the proposed guidewire tracking algorithm on three different endovascular videos. We show the original first image (left) and tracking result (right three images) for three consecutive frames. In each frame the contour is overlaid on top of the original image.

3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. Intl. Journal of Computer Vision **1**(4) (1987) 321–331
4. Casselles, V., Kimmel, R., Saprio, G.: Geodesic Active Contours. The Intl. Journal of Computer Vision **22**(1) (1997) 61–79
5. Cremers, D., Tischhauser, F., Weickert, J., Schnorr, C.: Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional. International Journal of Computer Vision **50**(3) (2002) 295–313
6. Brigger, P., Hoeg, J., Unser, M.: B-Spline Snakes: A Flexible Tool for Parametric Contour Detection. IEEE Trans. on Image Processing **9**(9) (2000) 1484–1496
7. Kovesi, P.: Image Features From Phase Congruency. Videre: A Journal of Computer Vision Research. MIT Press **1**(3) (1999)
8. Foley, J., van Dam, A., Feiner, S., Hughes, J.: Computer Graphics: Principles and Practice. Second edn. Addison-Wesley (1996)
9. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Trans. Image Process. **7**(3) (1998) 359–369