Pearling: Medical Image Segmentation with Pearl Strings

Jarek Rossignac[1], Brian Whited[1], Greg Slabaugh[2], Tong Fang[2], Gozde Unal[2]

[1] Georgia Institute of Technology
Graphics, Visualization, and Usability Center
Atlanta, GA 30332

[2]Intelligent Vision and Reasoning Department
Siemens Corporate Research
Princeton, NJ 08540

Abstract. This paper introduces a novel segmentation technique, called *pearling*, for identifying tubular structures in images. Examples of such structures include, but are not limited to, blood vessels, bones, roads, rivers, electrical wirings, and brush-strokes. Pearling allows the user to extract a higher level parametric representation of each tube or of a network of tubes. This representation is comprised of an ordered series of pearls, each defined by the location of its center, by its radius, and possibly by additional properties including a time value. A final model of the tube is then obtained by estimating continuous functions that interpolate the discrete series of pearls. Pearling is computationally efficient and well suited to user interactivity. We demonstrate its effectiveness in segmentation of medical images of different modalities as well as satellite imagery.

Key words: Vessel segmentation, anatomic modeling, interactive techniques

1 Introduction

Segmentation and geometric modeling of tubular structures such as blood vessels is a fundamental problem in medical imaging and is an important component of clinical applications involving diagnosis (e.g., stenosis, aneurysm, etc.) surgical planning, anatomic modeling and simulation, and treatment verification. The problem also arises in other contexts, including industrial applications as well as aerial/satellite image analysis.

With manual techniques, it is possible to obtain highly accurate segmentation results. However, these methods require too much tedious labor to be practical in a clinical workflow. While fully automatic segmentation methods can be desirable in these applications, given the poor contrast, noise, and clutter that is common to medical images, it is often difficult for fully automatic segmentation methods to robustly delineate the vasculature. Therefore, there is a salient need

for interactive segmentation methods that are *mostly* automatic, but do accept input from the user to correct for mis-segmentation or allow the user to guide the segmentation in a particular direction, the latter of which can be important when one is interested in a particular path through branching vasculature. Crucial to such a semi-automatic segmentation method is computational efficiency, so that the user will not wait an undue amount of time for segmentation results while interacting with the data.

Given its clinical importance, the problem of vessel segmentation has received a fair amount of attention in the literature. Kirbas et al. [1] present a recent survey that includes over 160 references, and proposes a taxonomy of six categories for grouping vessel segmentation methods, including pattern recognition techniques, model-based approaches, tracking-based approaches, techniques using artificial intelligence, neural networks, and tube-like detection algorithms. Some other recent papers on the subject include [2], which implements co-dimension two level set flows, [3], which solves a 4D minimal path problem representing the centerline and radius of the tube, [4], which implements level set flows with a tubular shape prior, and [5], which applies a Bayesian classifier to feature vectors produced using Gabor wavelets. However, nearly all of this previous work, with runtimes typically ranging from several seconds to minutes, is too slow for interactive segmentation.

In this paper we present *pearling*, which is a novel method for segmentation of tubular structures in images. Pearling performs a segmentation by computing an ordered series of *pearls*, which are variable-radius 2D disks. Starting with an initial pearl given by the user, as well as an initial direction, pearling computes the position and radius of an adjacent pearl based on the image data, so that the newly placed pearl fits properly inside the vessel. The method proceeds in this fashion, producing a string of pearls that provide a discrete representation of the vessel geometry. Pearling is robust to fluctuations in image intensities (due to noise, etc.) as the forces acting on a pearl are integrated over the region inside the pearl. A final smooth contour representing the tubular structure is then obtained by estimating continuous functions that interpolate the discrete series of pearls. As we will show, pearling is computationally efficient and well suited to user interactivity. This interactivity affords user guidance of the segmentation in a particular direction as well as user correction of errant segmentation results.

The rest of this paper is organized as follows. In Section 2 we describe our representation, segmentation approach, and estimation of a continuous contour from a discrete set of pearls. Then, in Section 3 we present some results showing the effectiveness of pearling, both for medical image segmentation as well as other applications like aerial imaging. Finally, we offer some concluding remarks in Section 4.

2 Methodology

Pearling allows the user to extract a higher level parametric representation of each tube or of a network of tubes. This representation is called a *string*. As

shown in Figure 1, it comprises an ordered series of pearls (or disks), each defined by the location $\mathbf{c}_i$ of its center, by its radius r_i, and a time value t_i, and possibly other attributes a_i. The model of the corresponding tube that is recovered by Pearling is defined as the region W swept by a pearl whose center $\mathbf{c}(t)$ and radius $r(t)$ are both continuous and smooth functions of the time parameter t. These functions interpolate the centers and radii of the string pearls for given values t_i of time. Although in this paper we present Pearling in 2D, the concept generalizes to 3D as well.

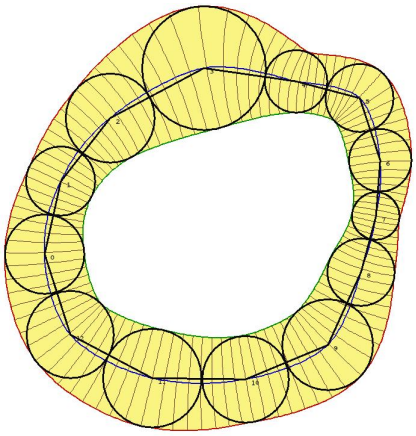


Fig. 1. Representation of Pearling, which consists of an ordered series of pearls. Continuous functions that interpolate the pearls are also shown: the blue curve interpolates the pearl centers, and the red and green curves interpolate the outside and inside edges of the pearls, respectively.

2.1 Estimation of pearls

Pearling starts with a first anchor, $\mathbf{c}_0$, which is a point either provided by an operator or by an algorithm. Typically, $\mathbf{c}_0$ should be chosen to lie close to the centerline of a tube and possibly close to one of the tube's ends. Pearling then uses an iterative process to construct an ordered series of pearls, one at a time. The center $\mathbf{c}_i$ and radius r_i of the current pearl is chosen so as to maximize r_i subject to image data, given the constraint that the distance d_i between $\mathbf{c}_i$ and the center $\mathbf{c}_{i-1}$ of the previous pearl is a given function $d(r_{i-1}, r_i)$ of r_{i-1} and r_i. We propose linear functions $d(r_{i-1}, r_i) = ar_{i-1} + br_i$, and in particular favor $d(r_{i-1}, r_i) = r_{i-1} + r_i$, where each pearl is tangent to the previous one or $d(r_{i-1}, r_i) = r_i$, where each pearl is tangent to the center of the previous pearl.

Let $\mathbf{c}_{i-1}$ be the current anchor, as shown in Figure 2. Let $\mathbf{c}_i$ be the center of the next pearl, and let r_i be its radius. The objective is to find the optimal

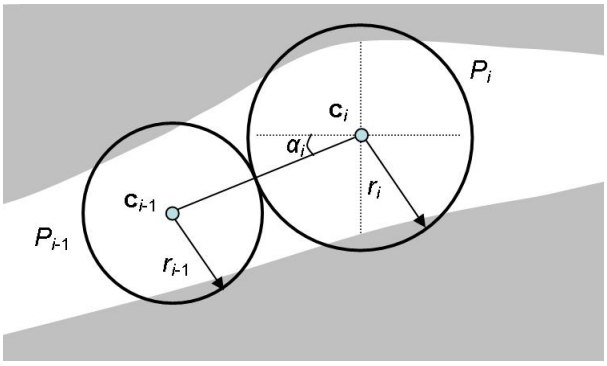


Fig. 2. Estimating the next pearl, P_i. See text for details.

values of $\mathbf{c}_i$ and r_i that place the ith pearl in the tube. The location of $\mathbf{c}_i$ may be specified in polar coordinates around $\mathbf{c}_{i-1}$. These comprise an orientation o_i and a radius r_i. The orientation may be defined in several ways, for example by a unit vector or by one angle (for example, α_i, as shown in the figure). In order to adjust the values of r_i and o_i so that the pearl better fits within the tube, we define two functions: $\mathbf{f}(\mathbf{c}_i, r_i)$ and $g(\mathbf{c}_i, r_i)$, as described below.

Orientation estimation. The function $\mathbf{f}(\mathbf{c}_i, r_i)$ returns a gradient indicating the direction in which $\mathbf{c}_i$ should be adjusted and the amount of the adjustment. This adjustment can then be converted to an angle relative to $\mathbf{c}_{i-1}$ and applied to o_i. $\mathbf{f}(\mathbf{c}_i, r_i)$ takes the form,

$$\mathbf{f}(\mathbf{c}_i, r_i) = \frac{1}{r_i^2} \int_{\mathbf{x} \in P_i} \phi(\mathbf{x})(\mathbf{c}_i - \mathbf{x}) \left(1 - \frac{\|\mathbf{c}_i - \mathbf{x}\|^2}{r_i^2}\right) d\mathbf{x} \tag{1}$$

$$\phi(\mathbf{x}) = \begin{cases} 1, & \text{if } \hat{p}_{\text{out}}(I(\mathbf{x})) > \hat{p}_{\text{in}}(I(\mathbf{x})) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The function $\mathbf{f}(\mathbf{c}_i, r_i)$ sums the vectors $(\mathbf{c}_i - \mathbf{x})$, where $\mathbf{x}$ is the vector coordinate of the the current pixel, across the entire area of pixels for pearl P_i, using only those pixels such that $\phi(\mathbf{x}) = 1$, i.e., pixels determined to be outside the tube. Each of these vectors is weighted by its distance from $\mathbf{c}_i$ such that pixels nearer $\mathbf{c}_i$ have a stronger influence on the result, as reflected in the $\left(1 - \frac{\|\mathbf{c}_i - \mathbf{x}\|^2}{r_i^2}\right)$ component of Equation 1. Intuitively, Equation 2 states that each point inside the ith pearl but outside the vessel imparts a force on the pearl that pushes it away from the vessel boundary. When the forces are balanced on all sides of the pearl, the pearl is typically centered in the vessel.

Determination of whether a pixel lies inside the tube or outside the tube is necessary when computing $f(\mathbf{c}_i, r_i)$, and is achieved using non-parametric density estimation. Before running the algorithm, the user selects two regions;

one inside the tube and one outside. For a given region, we estimate the density by applying a smoothing kernel K to the pixels in the region's histogram, i.e.,

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{I_i - m}{h}\right), \tag{3}$$

where I_i is the intensity of the ith pixel in the region, m is the mean of intensities of the n pixels in the region, and h is the bandwidth of the estimator. We use a Gaussian kernel, $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$. Performing this estimation on two user-supplied regions results in two densities, $\hat{p}_{\text{in}}(I)$ and $\hat{p}_{\text{out}}(I)$. During segmentation, an intensity I is classified as outside if $\hat{p}_{\text{out}}(I) > \hat{p}_{\text{in}}(I)$; otherwise it is classified as inside.

Radius estimation. In addition to adjusting o_i, r_i must also be adjusted to better fit inside the tube, which is done using the function $g(\mathbf{c}_i, r_i)$ as shown in Equation 4. For robustness, pearls are designed to have a percentage p of their pixels inside the tube and the rest outside the tube, as indicated in Figure 2. In our implementation, $g(\mathbf{c}_i, r_i)$ is positive when less than p percent of the pearl's pixels fit in the tube and negative more than $1-p$ of the pixels lie outside of the tube. The portion of the enlarged pearl that lies outside of the tube is typically computed by sampling the pearl's interior, however, alternatively one can sample the pearl's boundary. The result of $g(\mathbf{c}_i, r_i)$ is then used to scale r_i to better fit in the tube. $g(\mathbf{c}_i, r_i)$ takes the form,

$$g(\mathbf{c}_i, r_i) = p - \frac{\int_{\mathbf{x} \in P_i} (1 - \phi(\mathbf{x}))\, d\mathbf{x}}{\int_{\mathbf{x} \in P_i} d\mathbf{x}} \tag{4}$$

Interleaving the estimation and convergence. We interleave the estimation of o_i and r_i for the ith pearl P_i. For a given orientation o_i and radius r_i, one can update both parameters independently using the results given by $\mathbf{f}(r_i, \mathbf{c}_i)$ and $g(r_i, \mathbf{c}_i)$. In both cases, the quality of the fit is measured and the desired adjustment is computed and returned.

The adjustments of o_i and r_i are done through several iterations until the adjustment values returned by $\mathbf{f}(\mathbf{c}_i, r_i)$ and $g(\mathbf{c}_i, r_i)$ fall below a given threshold. The process then freezes the current pearl and starts fitting the next one. The whole process stops when the radius of the next pearl falls outside of a prescribed range, or when another application-dependent criterion is met. The result of this Pearling process is a series of location-radius pairs $(\mathbf{c}_i, r_i)$.

2.2 Building a continuous model

A continuous model W of the tube is obtained by computing continuous functions $\mathbf{c}(t)$ and $r(t)$ that interpolate the discrete set of pearls. We model W as the union of an infinite set of balls [6], which is and may be expressed in parametric form as $W = \text{Disk}(\mathbf{c}(t), r(t))$ for $t \in [0, 1]$. If we assume that the pearls are more

or less uniformly spaced along the canal, we can define the canal as the limit of a four-point subdivision process, which, at each iteration, introduces a new pearl between each pair of consecutive pearls. It computes the center position and radius of the new pearl as a weighted sum of the corresponding parameters of the neighboring pearls before the subdivision. For example, consider any original sequence of four consecutive pearls: $\cdots P_a, P_b, P_c, P_d \cdots$. It will be refined into $\cdots P_a, P_l, P_b, P_m, P_c, P_n, P_d \cdots$. In particular, each parameter q (say x, y, r) of P_m may be computed from the corresponding parameters of P_a, P_b, P_c, and P_d using

$$q(P_m) = \frac{-q(P_a) + 9q(P_b) + 9q(P_c) - q(P_d)}{16}. \tag{5}$$

Note that the formulation of the four-point interpolation corresponds to a local cubic fit. It is obtained by solving for the coefficient $d = F(0)$ of a cubic function $F(u) = au^3 + bu^2 + cu + d$, given the constraints are: $F(-3) = q(P_a), F(-1) = q(P_b), F(1) = q(P_c)$, and $F(3) = q(P_d)$. Strings for which the beginning pearl is not equal to the ending pearl are extended by two pearls so that the subdivision has enough region of support to compute Equation 5 at the end pearls.

Since the values of r_i are larger than the actual tube, as explained in the previous section, the values must be adjusted to better fit. The radii are chosen to have a percentage p of pixels inside the tube, and the rest outside. If we multiply all values of r_i by p as used in computing $g(\mathbf{c}_i, r_i)$, we get the true radii which will be used in the continuous model W.

Figure 3 provides an example of an original string of 6 pearls and the results of 1, 2, and 5 subdivisions. The original string may have been obtained through a decimation of the string constructed by the pearling algorithm presented here.

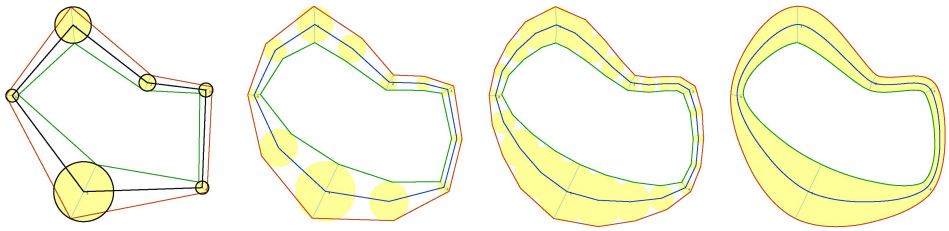


Fig. 3. Subdivision for estimating the continuous canal surface that interpolates the pearls.

The result of several subdivision steps may be compared to the actual data. If a span between two consecutive pearls of the original string does not accurately fit to the data, an additional pearl may be introduced in the middle point of the segment. This pearl is chosen from the initial string, before decimation. We show below an original control string of 6 pearls and its canal (left) and a new canal produced by a string where a new pearl has been added (right). Note that spans

to the left and right of the span where the new pearl was inserted have now been modified and need to be checked again for accuracy, and possibly subdivided.

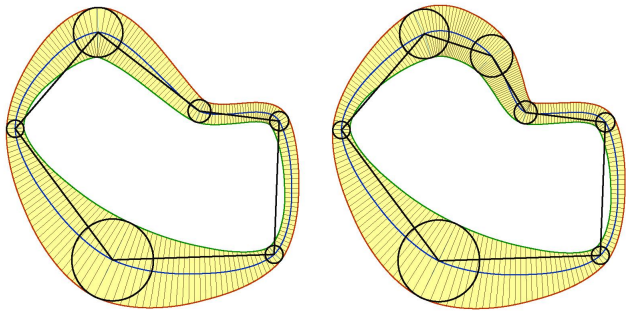


Fig. 4. The effect on the canal of adding a pearl to a decimated string.

The red and green curves, which bound the canal on each side are each generated as a polyline that interpolates one point on each circle. The point is chosen at a place on the circle where the normal is orthogonal to the corresponding tangent line to the previous and next circle. The polylines are shown below for different levels of four-point subdivision.

3 Results

We now present segmentation results using Pearling. We begin with the segmentation of an ultrasound image of a carotid artery; the original image is shown in Figure 5 (a). As indicated in (b) of the figure, the user selects two regions (shown in red and green), which are used to estimate the non-parametric densities described in Section 2.1. Then, the user selects a starting anchor pearl an initial direction (shown in blue in (b)). The algorithm then proceeds, successively adding pearls until no more can be added, as shown in (c) of the figure. From the collection of discrete pearls, we then extract the continuous model, producing the final segmentation shown in (d). Using Pearling, this 655x373 image was segmented in 300 ms, with an average of 5 iterations of orientation and radius estimation per pearl.

Figures 6, 7, and 8 present more segmentation results, of an MR angiographic image, and x-ray angiographic image, and a satellite image of a subdivision. For each figure, in (a) we show the regions (depicted with a red and green color) used for density estimation, as well as the initial anchor pearl and its initial direction. In (b) of each figure we present the final segmentation result. Table 1 shows the image size, segmentation time, and number of iterations per pearl for these images. Note that all completed in a few hundred milliseconds.

For comparison, we applied non-parametric level set segmentation to the same images using the same estimated densities. Our level set segmentation application is implemented in C++ using efficient narrowband techniques [7]. Results are presented in Figure 9. The carotid image (shown in (a)) was segmented in 10.2 seconds, which is roughly two orders of magnitude slower than Pearling, but achieves a comparable result in terms of quality. The MR angiography image (shown in (b)) took 34.5 seconds, and the satellite image (shown in (c)) took over 2 minutes to complete. In both (b) and (c), the level set method automatically follows all branches through the network. In cases where one is only interested in an individual branch, this results in unnecessary computations that can cause delays when presenting results to the user.

Data Set	Image size	Segmentation time	Iterations per pearl
Carotid	655 x 373	300 ms	5
MR angiography	350 x 598	170 ms	10
X-ray angiography	870 x 724	162 ms	12
Subdivision	1001 x 634	340 ms	6

Table 1. Image size and runtimes for the segmentations using Pearling shown in Figures 5, 6, 7, and 8.

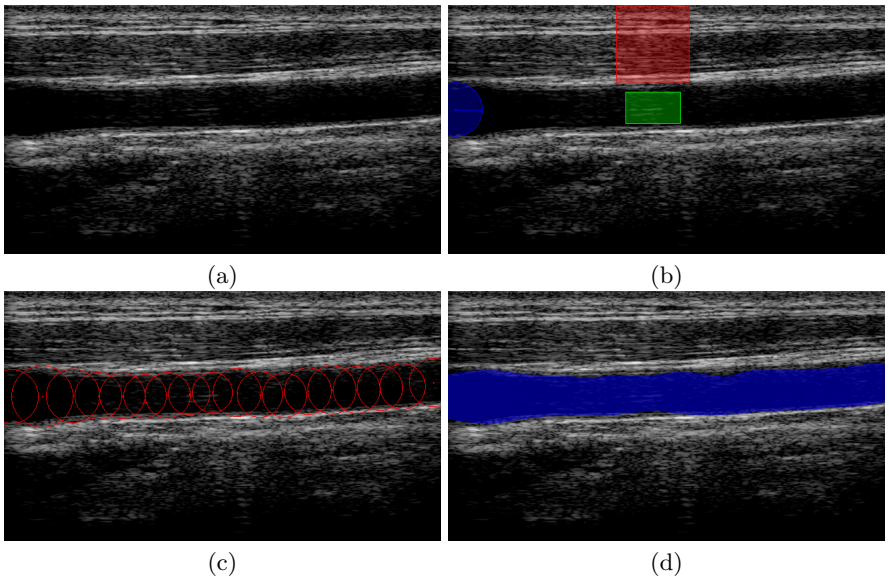
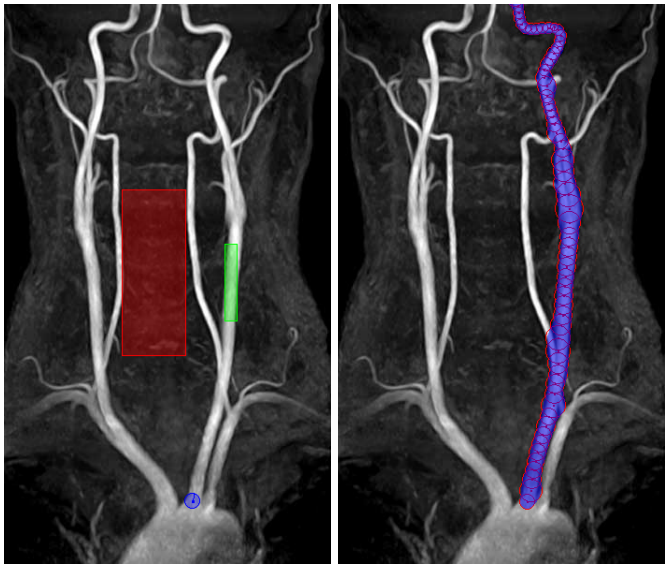


Fig. 5. Segmentation of a carotid artery imaged using ultrasound.



(a) (b)

Fig. 6. Segmentation of an MR angiographic image

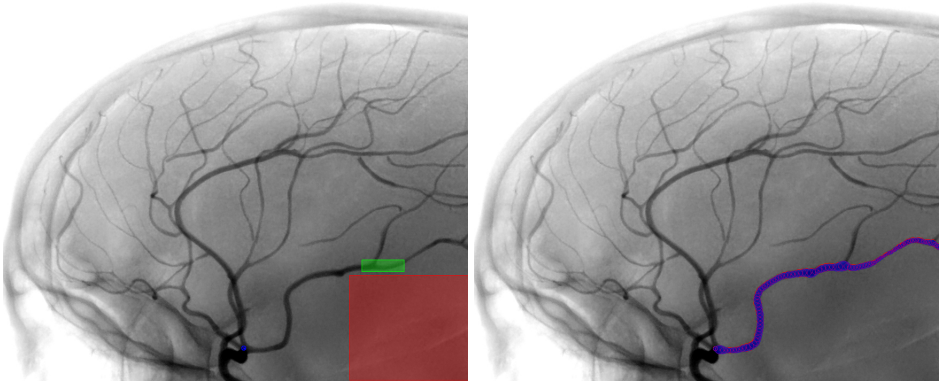
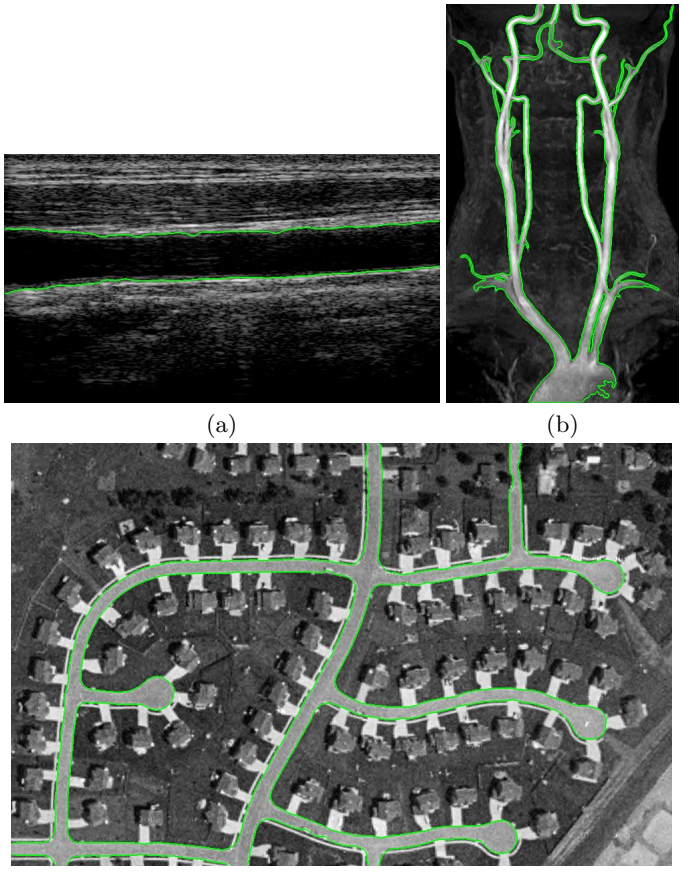


Fig. 7. Segmentation of an x-ray angiographic image.

3.1 User interactivity

As Pearling is computationally efficient, it affords user interaction with data without delays due to computational complexity. In Figure 10 we show an ex-

Fig. 8. Segmentation of a satellite image of a subdivision.



(a) (b)

(c)

Fig. 9. Comparison with non-parametric level sets segmentation.

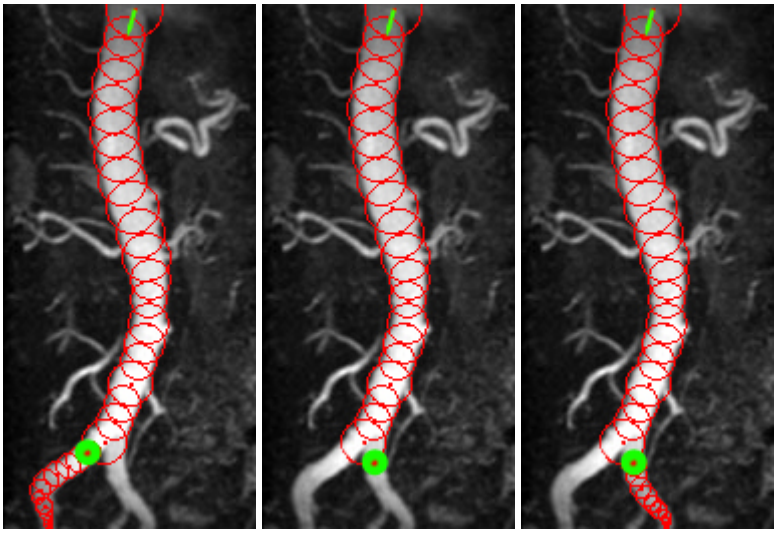
ample where a user can add a branch to an existing segmentation with two mouse motions, by simply selecting the pearl where the branch should occur and an initial direction. The new branch (shown in green) is computed quickly (in a few hundred milliseconds) and is display alongside the previous segmentation (shown in blue).

In Figure 11, we show how a user can choose a different branch through bifurcating vasculature. In this example, the segmentation initially takes the left branch of a bifurcating vessel. In this case, however, the user would prefer the segmentation of the right vessel. The user selects a pearl S (shown in bright green in (a)) after which all subsequent pearls in the chain are deleted. Then, in (b), the user moves S to a new position in the right vessel. Upon releasing the pearl, Pearling segments the new branch.

Fig. 10. Adding a new branch to an existing segmentation.

4 Conclusion

In this paper we presented Pearling, a new method for semi-automatic segmentation and geometric modeling of tubular structures. Pearling performs a segmentation by computing an ordered series of pearls that discretely model the tubular structure geometry. Smooth curves are then used continuously interpolate the pearls. The computational efficiency of Pearling affords efficient user



(a) (b) (c)

Fig. 11. Selecting a different branch in bifurcating vasculature.

interaction with the segmentation, allowing the user to correct for errant segmentation results or guide the segmentation in a particular direction through the data. While more comprehensive clinical validation of the algorithm is required, from our experimental results we conclude that Pearling results in highly efficient segmentation of tubular structures, and holds much promise for semi-automatic image segmentation.

References

1. Kirbas, C., Quek, F.: A Review of Vessel Extraction Techniques and Algorithms. ACM Computing Surveys **36**(2) (2004) 81–121
2. Lorigo, L.M., Faugeras, O.D., Grimson, W.E.L., Keriven, R., Kikinis, R., Nabavi, A., Westin, C.F.: CURVES: Curve Evolution for Vessel Segmentation. Medical Image Analysis **5** (2001) 195–206
3. Li, H., Yezzi, A.: Vessels as 4D Curves: Global Minimal 4D Paths to 3D Tubular Structure Extraction. In: Workshop on Mathematical Methods in Biomedical Image Analysis. (2006)
4. Nain, D., Yezzi, T., Turk, G.: Vessel Segmentation Using a Shape Driven Flow. In: MICCAI. (2004)
5. Soares, J., Leandro, J., Cesar, R., Jelinek, H., Cree, M.: Retinal Vessel Segmentation Using the 2-D Gabor Wavelet and Supervised Classification. IEEE Trans. on Med. Img. **25**(9) (2006) 1214–1222
6. Monge, G.: Applications de l'analyse à la géométrie. Fifth edn. Bachelier, Paris (1894)

7. Sethian, J.A.: Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press (1999)