

Semi-Automated Extraction of Canine Left Ventricular Purkinje Fiber Network

Jie Li¹, Kuanquan Wang¹, Wangmeng Zuo¹, Henggui Zhang^{1,2}

¹School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

²School of Physics and Astronomy, University of Manchester, Manchester, UK

Abstract

The purkinje fiber network (PFN) is a very important conduction system in the endocardial surface of the ventricle, whose structure is crucial in ventricular physiopathology. Traditional medical imaging methods, such as magnetic resonance imaging (MRI) or computed tomography (CT), however, fail to reveal the detailed PFN information. An alternative is to model PFN as idealized self-similar fractal tree. Recently, a LLE-based method is proposed for the construction of the purkinje system in the canine left ventricle (LV), where curvilinear PFN structures are first detected from 2D image and then mapped to 3D surface. This method, however, adopts a simple local thresholding method to extract the curvilinear PFN structure, and thus many interactions are required to obtain the satisfactory detection result. In this work, we propose a semi-automated method for extracting both the location and the width information from the dissection image of the endocardial surface of the canine left ventricle, which is more feasible and adaptive for curvilinear PFN structure extraction.

1. Introduction

The coordinated contraction of the heart is regulated by the specialized electrical conduction system of the heart, which consists of the sino-atrial node, Bachmann's bundle, and the purkinje fiber network (PFN) etc [1, 2]. PFN, which includes the bundle branch and the purkinje fiber, is crucial in modeling ventricular pathophysiological mechanisms.

Purkinje fiber is generally confined to the endocardial surface, which makes the imaging of the 3D PFN very difficult. Recently, Li et al. proposed a LLE-based method to construct the canine left ventricular PFN by using the dissection image of the endocardial surface of the canine left ventricle [3]. However, the structure of PFN is complicated and the color of the purkinje fiber is hardly distinguished from that of the background. Since Li et al.'s method only adopted a simple local thresholding method to extract the curvilinear PFN structure, many interactions are required to obtain the satisfactory detection result.

In this work, we intend to utilize the progress in automated or semi-automated curvilinear structure detection methods to make the PFN detection more feasible. Recently, several classes of methods have been proposed to segment vessels in medical images, which generally rely on the use of a local detector and a post-processing method to link the locally detected vessels [4].

The notion of the shortest or geodesic path, which is proved to be efficient for the extraction of salient curves in 2D or 3D images, is shown in [5, 6]. Geodesic curves can also be used to extract 3D tubular structure centerlines, as proposed by Deschamps and Cohen [7] and by Santamaria et al. [8]. In [4], Péchaud et al. developed a four-dimensional fast marching method to extract the centerlines of the vessel network together with the local orientation and scale information.

Steger [9] proposed a model for the unbiased detection of curvilinear structures with subpixel accuracy. This method can obtain not only the precise subpixel line position but also the width of the line for each line point.

In our study, we propose a novel semi-automated PFN extraction method. By combining the fast marching algorithm [10] and Steger's line width detector [9], the proposed method can simultaneously obtain the location and the width information of the curvilinear PFN structure.

The remainder of the paper is organized as follows: Section 2 proposes materials and methods; Section 3 provides results of the application; Finally, concluding remarks are given in Section 4.

2. Materials and methods

2.1. Materials

We use the dissection image of the endocardial surface of the canine left ventricle, which is provided by Dr. Cherry and Dr. Fenton of Cornell University. The dissection image is shown in Figure 1.

2.2. Methods

Since the whole PFN is composed of many curvilinear

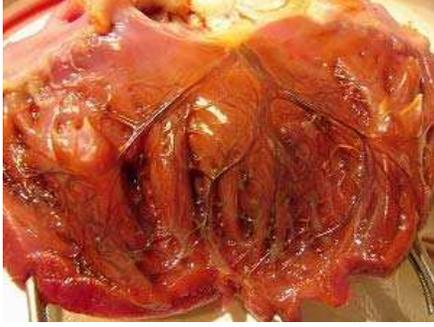


Figure 1. The image of the PFN

structures, we have to deal with each curvilinear structure firstly, then perform the extraction in the whole PFN. For one curvilinear branch, the method makes use of user input only for obtaining seed points of several curves in PFN. Then the centerline of one curve is found by the fast marching method of Sethian [10]; next, we refine points of the centerline, and detect lines with the extraction of edge in the theory of curvilinear structures by Carsten Steger [9]. For the next branch, besides the above processing, it has to judge the crossing part. After all the curves are detected, finally we will obtain profiles of the PFN. Figure 2. is the sketch of the method.

Detailed algorithm is introduced as the following:

- Step 1—Initialization: marking m seed points $\{SP_k | 1 \leq k \leq m\}$ of one curvilinear branch on the image manually.
- Step 2— Judging the value of k , If $k < m$, then continue; or else, jump to Step 5.

For each couple of adjacent points (SP_k, SP_{k+1}), computing the approximate centerlines $\{P_i | 1 \leq i \leq n\}$ of the PFN curvilinear structure between SP_k and SP_{k+1} . The process will be explained in Section 2.2.1.

Purifying approximate centerlines, as Section 2.2.2 explains.

- Step 3— Computing the possible edge points for P_i and P_{i+1} , as is explained in Section 2.2.3

Purifying the possible edge points to the group of reasonable edge points as is explained in Section 2.2.4.

- Step 4— Determining the junction in each side of edge points of P_i and P_{i+1} , as is explained in Section 2.2.5;

$k+1 \rightarrow k$, go to Step 2.

- Step 5— Detection of points in crossing parts of curves and the isolated points, as is explained in Section 2.2.6.

- Step 6— Providing users choices, whether they will continue to input seed points or not. If they will, then go to Step 1, or else, the algorithm go to the end.

2.2.1. Computing approximate centerlines

Here we use the classical fast marching method by Sethian [10].

We will find approximate centerline points by fast marching method; however, there are many points among

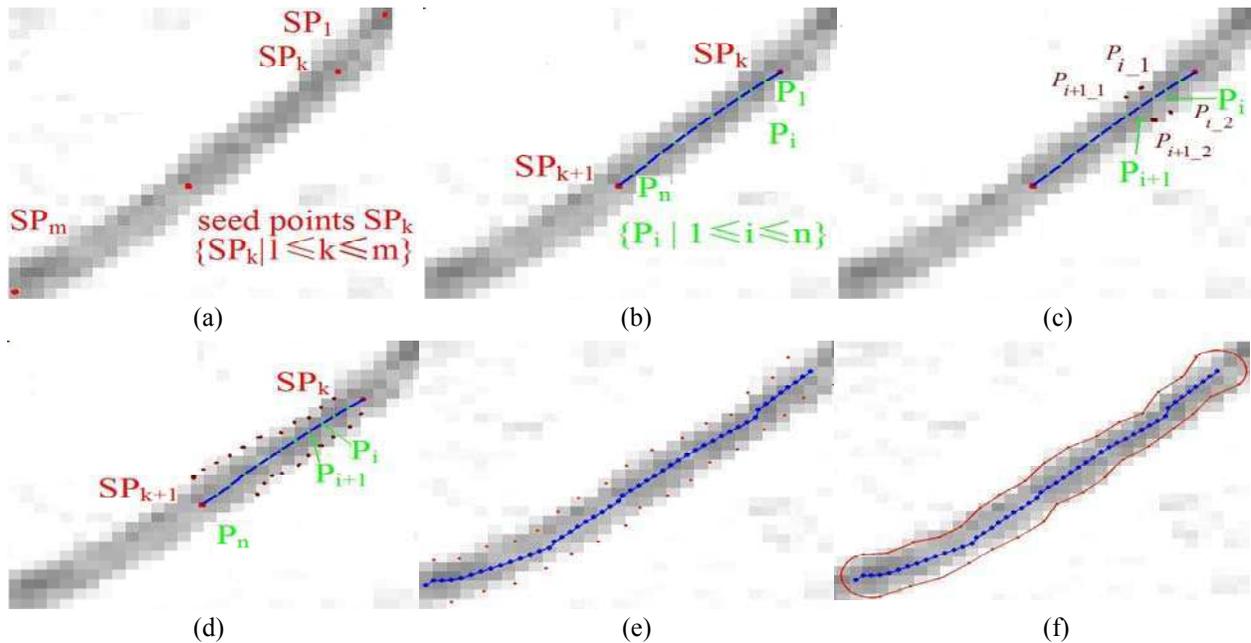


Figure 2. Sketch of the method. (a) Users input seed points (b) Generating the centerline (c) For each couple of adjacent points of the centerline, to find the edge points (d) Finding edge points of points in the centerline between two seed points (e) Finding edge points of the whole curve (f) the profile of the curve

them useless, which we have to remove, to reduce the complexity of the computation:

- a. Several points in one pixel: number of the points won't increase precision, thus it is enough to take one point in one pixel;
- b. Two close points in adjacent pixels: when distance between two points is less than one pixel, it also is enough to take either point.

2.2.2. Purifying approximate centerlines

Following Section 2.2.1, we can obtain a set of centerline points. We take half of the points uniformly, with seed points and special points are preserved. Here, 'special points' is referred to points at which the local centerline has a large gradient.

2.2.3. Computing edge points of one centerline point

The cubic spline is generated by the approximate centerline from the beginning seed point to the ending seed point. For the i th point P_i , let \bar{n} is the normal of the spline at P_i , take the range of 2α which is divided by \bar{n} as the searching area candidate, then take another square area with the center point P_i and length of side $2a$ as another searching area candidate. The area in both two searching area candidates is effective searching areas, as shallows in Figure 3.

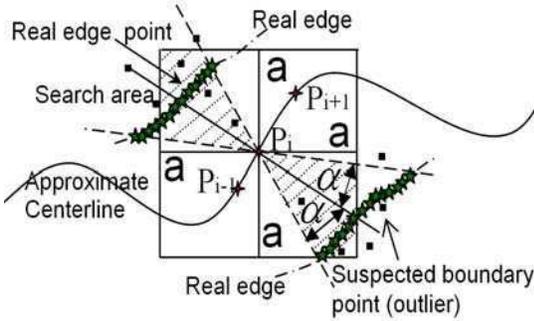


Figure 3. Sketch of the searching areas of P_i

Then we will search the gray value in the effective searching areas, to detect appropriate points as the edge of the local centerline corresponding to the point P_i . The process is similar to [9].

2.2.4. Purifying the possible edge points

For one point of the centerline, it may have several appropriated points, but in each side, only one of them is the edge point, so we always pick the one with maximum value of all the appropriate points, and verify whether it is the real appropriate point, as is introduced in Section 2.1.3. If it is, then it is the edge point of P_i , or else, it will be thrown away, and continue the purifying in left points.

To be important, it is necessary to judge whether the appropriate point is an isolated edge point. If it is, then the point should be thrown away.

Then the edge points of P_i in centerline are found, we use P_{i-1} , P_{i+1} stand for each edge point of P_i .

2.2.5. Determining the junction in each side of edge points of P_i and P_{i+1}

Here, we mark points as: $P_i(x_i, y_i)$, $P_{i+1}(x_{i+1}, y_{i+1})$; the edge point in each side of P_i is (x_i^*, y_i^*) , and of P_{i+1} is (x_{i+1}^*, y_{i+1}^*) . For P_i and P_{i+1} , let 'left side' stands for the left side of \bar{n} along its direction; 'right side' stands for the right side of \bar{n} along its direction; and β is the slope of the centerline at the point P_i or P_{i+1} .

For P_i , (x_i^*, y_i^*) is the left side edge point, \Leftrightarrow

$$\left\{ \begin{array}{l} \beta > 0 \& \& \left\{ \begin{array}{l} ((y_{i+1} - y_i) > 0 \& \& (y_i^* - \beta x_i^*) > 0) \\ ((y_{i+1} - y_i) < 0 \& \& (y_i^* - \beta x_i^*) < 0) \end{array} \right. \\ \parallel \\ \beta < 0 \& \& \left\{ \begin{array}{l} ((y_{i+1} - y_i) > 0 \& \& (y_i^* - \beta x_i^*) < 0) \\ ((y_{i+1} - y_i) < 0 \& \& (y_i^* - \beta x_i^*) > 0) \end{array} \right. \end{array} \right.$$

Thus, we can differentiate edge points of P_i in both side, also we can find the same sides of edge points. At last, we attach P_{i-1} with P_{i+1} , P_{i-2} with P_{i+2} .

2.2.6. Detection of points in crossing parts and the isolated points

After detecting two curves, if there are any isolated points, then remove them; and the method will hint users to judge whether there are crossing parts. If there are, then edge points are searched to detect points in crossing parts, or else, then it will go to the end of this step.

3. Results

Figure 4 shows a part of one curvilinear extracted from the image. To be clear to seen, the image is one section of the whole PFN. Points A, B are seeding points, P_i stands for points of the centerline.

Figure 5 shows another curve, which is a whole curvilinear in the network. It is extracted by the input seeding points C, D, E.

Figure 6 shows a curvilinear structure with the crossing part of the whole PFN in the image.



Figure 4. A single curveline and its branches extraction



Figure 5. The extraction of the network

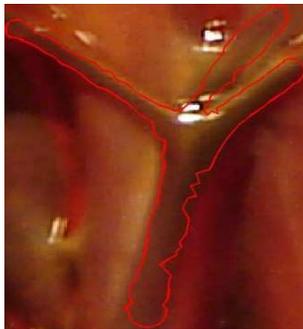


Figure 6. The crossing part of the extraction

4. Conclusion

In this paper, we propose a novel semi-automated PFN extraction method for the extraction of the PFN structure in the endocardial surface of the canine left ventricle. By combining the fast marching algorithm [10] and Steger's line width detector [9], the proposed method is effective for extracting PFN from the background with the similar gray scale to the foreground. Moreover, our method can directly and naturally compute both the centerlines and the edge points of the fibers. With this method, a semi-automated scheme can be used to extract PFN conveniently by only manually providing the starting and ending points of each Purkinje fibers.

Acknowledgements

This work is supported by the National High-tech Research and Development Program of China ("863" Program) under Grant No. 2006AA01Z308 and the National Natural Science Foundation of China (NSFC) under Grant No. 60872099.

The authors would like to thank Dr. Cherry and Dr. Fenton of Cornell University for providing the dissection image of anatomical canine left ventricle, and thank Dr. Peyre of Universite Paris-Dauphine for making the Toolbox Fast Marching public available.

References

- [1] Mikawa T, Hurtado R. Development of the cardiac conduction system. *Seminars in Cell & Developmental Biology* 2007;18: 90–100.
- [2] Silverman ME, Grove D, Upshaw CB. Why does the heart beat? The discovery of the electrical system of the heart. *Circulation* 2006;113: 2775–2781.
- [3] Li J, Wang KQ, Zuo WM, Yuan YF, Zhang HG. Canine Left Ventricular Purkinje Fiber Network Construction Using Manifold Learning. *Computers in Cardiology* 2009;36:465–468.
- [4] Péchaud M, Peyré G, Keriven R. Extraction of Vessels Networks over an Orientation Domain. [http://hal.archives-ouvertes.fr/hal-00365021/ fr/](http://hal.archives-ouvertes.fr/hal-00365021/fr/). Version1-2, Mar 2009
- [5] Cohen L, Kimmel R. Fast marching the global minimum of active contours. *Proc. IEEE International Conference on Image Processing 1996; (ICIP'96)*.
- [6] Poon K, Hamarneh G, Abugharbieh G. Live-vessel. Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images. In *MICCAI 2007; 2: 444–451*.
- [7] Deschamps T, Cohen L. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis* 2001; 5: 281–299.
- [8] Santamaría-Pang A, Colbert CM, Saggau P, et al. Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging. In *MICCAI 2007; 2:486–494*.
- [9] Steger C. An Unbiased Detector of Curvilinear Structures. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 1998; 20: 113–125.
- [10] Sethian J. Fast marching methods. *SIAM Rev* 1999; 41(2): 199–235.

Address for correspondence

Kuanquan Wang
 Mailbox 332, Harbin Institute of Technology
 Harbin 150001, China
 E-mail: wangkq@hit.edu.cn