Detection of Curves with Unknown Endpoints using Minimal Path Techniques

Vivek Kaul[1]
gtg819r@mail.gatech.edu
Yichang(James) Tsai[2]
james.tsai@ce.gatech.edu
Anthony Yezzi[1]
anthony.yezzi@ece.gatech.edu

[1] Georgia Institute of Technology
Atlanta
GA, USA

[2] Georgia Institute of Technology
Savannah,
GA, USA

Abstract

We present a novel method to detect curves with unknown endpoints using minimal path techniques. Our work builds on the state of the art minimal path techniques currently used to detect curves. Existing algorithms in the literature require the user to specify both endpoints of the curve or one endpoint plus the total length of the curve. However, in our approach, the user may specify any arbitrary initial point on the curve and the algorithm can detect the complete curve (even with multiple branches) automatically without the need for any additional information. We apply this algorithm to the problem of crack detection in civil structures where cracks are modeled as 2D curves. The results demonstrate that the algorithm is robust to variations in background and texture and is able to detect curves accurately.

1 Introduction

Minimal path techniques have been used to detect features in images that can be modeled as curves. The current minimal path theory works only with prior knowledge about both the endpoints or one end point plus the total length of the open curve. We propose a novel algorithm that relaxes the user input requirements of existing techniques and detects the complete curve (even with branches) assuming the knowledge of only one arbitrary point on the curve. This algorithm is applied to detect features that can be modeled as open curves: cracks in structures and narrow elongated features in medical images. This procedure can also be extended to closed curves and more complex topologies consisting of both closed curves and open curves.

2 Prior Work and Background

Many problems in computer vision and image processing have been tackled using variational energy minimization techniques. Terzopoulos et al. [9] proposed the active contour

BMVC 2010 doi:10.5244/C.24.62

model (snakes) that has been developed further for various applications like image segmentation, feature extraction and tracking. An active contour is a curve that deforms its shape to minimize an energy composed of two components: an internal energy component that smoothes the curve, and an external energy component that guides the curve towards desired image features. In active contour models, the curve initializations have to be performed very carefully, otherwise the final evolved curves may become trapped within local minima rather than the desired objects or features. To overcome some of the shortcomings of active contour models, Cohen and Kimmel [5] proposed a minimal path approach that captures the global minimizer curve of the contour dependent energy between two user defined endpoints. The image dependent energy is selected so that it takes lower values at features of interest: for instance, an energy directly dependent on the intensity value at each pixel can be chosen for detecting cracks (as cracks typically appear darker than their immediate surroundings). The original minimal path technique can be extended for 3D tree-structured object extraction [7], but not for general 3D surface extraction. Ardon et al. [1] proposed a more general scheme for 3D surface extraction between user supplied end-curves, but this scheme also requires the user supplied curves to be located precisely on the desired 3D boundary. Minimal path techniques have been applied to various applications like feature detection, contour completion [4], tubular surface extraction [7] and motion tracking [3].

All the above approaches require precise knowledge of endpoints (2D) or end curves (3D). In [2], a variant of the minimal path approach called Minimal Path Method With KeyPoint Detection (MPWKD) was devised to find curves under less restrictive prior information. The user needs to provide one endpoint on the curve that will lead to detection of representative keypoints along the curve using front propagation. For closed curves, a stopping criterion was derived that requires no further prior information from the user. Other researchers have used statistical shape priors [8, 16] and Principal Component Analysis (PCA) [11] based initial conditions for extracting organ shapes like lungs and kidneys as closed contours. However, for open curves, information about either both the endpoints or one endpoint plus the total length of the curve is required to find the complete curve [2]. If there are multiple branches in a curve then all the endpoints need to be known up front. The motivation of the current work is to find curves with less restrictive prior knowledge. In particular, the current algorithm requires the user to supply just one arbitrary point (not necessarily an endpoint) on the desired curve. The algorithm can find the complete curve (including endpoints and branches) only using this information. This work will be very useful in the detection of cracks in structures and detection of other features like optical nerves and bone cracks that can be modeled as curves.

3 Methodology

3.1 Minimal Path Theory

Given a 2D or 3D image $I : \Omega \rightarrow \mathbb{R}^+$ and two points p_1 and p_2, the idea introduced by Cohen and Kimmel [5] is to build a potential $\Phi : \Omega \rightarrow \mathbb{R}^+$, where $\Phi > 0$, that takes lower values near the desired features of the image I. The choice of the potential Φ depends on the application. For example, the potential function Φ for cracks can be taken to be a function of intensity value at each pixel because cracks are darker than the background. In some other applications, edge based potential functions can be used. The goal is to extract a curve that

minimizes the energy functional $E : \mathscr{A}_{p_1,p_2} \rightarrow \mathbb{R}^+$

$$E(\gamma) = \int_{\gamma} \Phi(\gamma(s))\mathrm{d}s, \tag{1}$$

where $\mathscr{A}_{p_1,p_2}$ is the set of all paths connecting p_1 to p_2 and s is the arc length parameter. The curve connecting p_1 to p_2 that globally minimizes the energy $E(\gamma)$ is called the *minimal path* between p_1 and p_2 or the geodesic. The geodesic curve is denoted by C_{p_1,p_2}. The solution of this minimization problem is obtained through the computation of the minimal action map $U_1 : \Omega \rightarrow \mathbb{R}^+$ associated with p_1. The minimal action map is defined as the minimal energy integrated along a path between p_1 and any point x of the domain Ω:

$$\forall x \in \Omega, U_1(x) = \min_{\gamma \in \mathscr{A}_{p_1,x}} \int_{\gamma} \Phi(\gamma(s))\mathrm{d}s. \tag{2}$$

In this document, the minimal action map will also be called the geodesic distance map. The values of U_1 are the arrival times of a front propagating from the source p_1 with velocity $(1/\Phi)$ and $U_1(p_1) = 0$. U_1 satisfies the Eikonal equation:

$$||\nabla U_1(x)|| = \Phi(x) \quad \text{for} \quad x \in \Omega. \tag{3}$$

The minimal action map calculation can be generalized to the case of multiple sources. The minimal action map or the geodesic distance map associated with the potential $\Phi : \Omega \rightarrow \mathbb{R}^+$ with a set of n sources $S = p_1, \ldots, p_n$ is the function $U : \Omega \rightarrow \mathbb{R}^+$, where

$$\forall x \in \Omega, U(x) = \min_{1 \leq j \leq n} \{U_j(x)\}, \tag{4}$$

and

$$U_j(x) = \min_{\gamma \in \mathscr{A}_{p_j,x}} \int_{\gamma} \Phi(\gamma(s))\mathrm{d}s. \tag{5}$$

$U_j(x)$ is given by Equation 2 where p_1 is replaced by p_j. For $p_j \in S$, $U(p_j) = 0$ and U satisfies the Eikonal equation given by Equation 3.

Another important quantity for the current research is the Euclidean distance map. It is the function $L : \Omega \rightarrow \mathbb{R}^+$ that assigns the Euclidean length of the minimal geodesic between x and the source S to every point x of the domain Ω.

$$\forall x \in \Omega, L(x) = \int_{C_{p_j,x}} \mathrm{d}s, \tag{6}$$

where

$$C_{p_j,x} = \min_{1 \leq i \leq n} C_{p_i,x}$$

is the minimum of all minimal paths from x to each source point p_i. $C_{p_i,x}$ is calculated using the potential Φ. If $\Phi = 1$ for all $x \in \Omega$, then the geodesic distance map U coincides with the Euclidean distance map L.

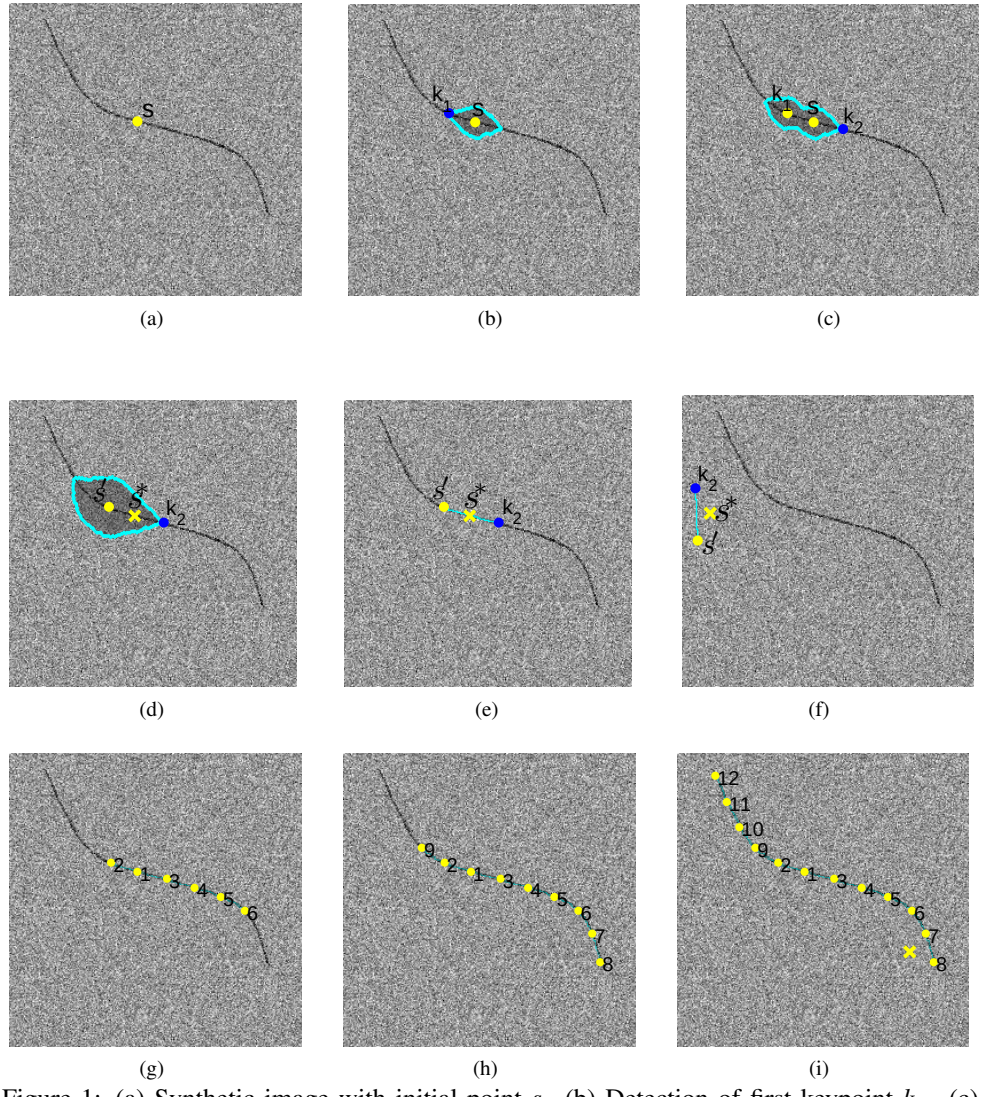
Sethian [12] proposed the Fast Marching method to solve the Eikonal equation that relies on a one-sided derivative that looks in the up-wind direction of the moving front. An alternate approach to solve the Eikonal equation was provided by Tsitsiklis in [14]. Given the source points, the Fast Marching algorithm helps to calculate geodesic distance map U and the Euclidean distance map L for every grid point. The details of the algorithm are given in

[2]. The computation of the minimal action map U for a single source p_1 is done according to the discretization scheme for Fast Marching given in [12]. The minimum of U is at the front propagation start point p_1, where $U(p_1) = 0$. The gradient of U is orthogonal to the propagating fronts since these are its level sets. Therefore, the minimal path between any point q and the start point is found by sliding back along the gradient of the map U until arriving at $U(p_1)$. This back propagation procedure is a simple steepest gradient descent. It is possible to make a straight forward implementation on a rectangular grid: given a point q, the next point in the chain connecting q to p_1 is selected to be the neighbor p' of q for which $U(p')$ is minimum. The back propagation approach can be easily extended to the multiple source points $(p_1, p_2, \ldots, p_n)$ scenario. In the case of multiple source points, the back propagation procedure gives the minimal path between point q and the set of source points $S = p_1, p_2, \ldots, p_n$. In addition, we can also find the end point $s^* \in S$ from which the minimal path to q originates. The minimal path back propagation procedure will be important for our algorithm and we will denote it by $MinimalPathbk(S, q)$, where S is the set of source points and q is the destination point.

The minimal path procedure described above requires the knowledge of two endpoints. In addition, it also assumes that the potential function is not very noisy and provides enough contrast that can enable the minimal path to track even convoluted, long curves along desired features. In many applications, these conditions are not met and the desired feature points have a lower potential only in a local neighborhood region. To overcome the limitations of this approach, Benhamasour and Cohen [2] introduced the Minimal Path Method With KeyPoint Detection (MPWKD) approach. This algorithm is based on the idea that among all points on the Fast Marching boundary that have equal geodesic distance (U) values, the points near the desired features will have the maximum Euclidean distance L. As the potential Φ at the feature points is lower than the neighborhood points, the Fast Marching boundary propagates with high speed $1/\Phi$ and travels the furthest Euclidean distance along the feature points. When the Fast Marching boundary propagates with the potential Φ from a set S, the first point for which Euclidean distance L exceeds λ is identified. This point is referred to as a *keypoint*. Keypoints are recursively detected until a known endpoint is reached or until the detected curve exceeds a given Euclidean length. The MPWKD requires prior knowledge of both endpoints or one endpoint plus the total length of the curve. If the curve has multiple branches then the user needs to know all endpoints of the curve. Our proposed algorithm addresses this concern and detects curves with multiple branches just with a user supplied arbitrary point on the curve. We discuss the algorithm next.

3.2 Curve Detection Algorithm

The *keypoints* detected in the MPWKD are approximately equally spaced along the curve. We use this fact to find a good stopping criterion for curve endpoint detection. Starting from the initial point s, fronts are propagated with potential Φ and the first point that crosses a Euclidean distance of λ is detected. This point is labeled as the first keypoint k_1. The Fast Marching procedure to find the first keypoint k_1 is referred as $FMM(S, \lambda)$ in this document where S is the source point set. The source point set S includes only the point s initially. We use a synthetic image that has a curve with lower mean intensity value than the random background for illustrating the algorithm. A λ value of 30 was used for this image and the potential function was chosen to be the intensity value. Figure 1a shows the image with initial source point s on the curve. Figure 1b illustrates the use of $FMM(S, \lambda)$ to find k_1. The points inside the Fast Marching boundary are scaled to a lower intensity value for better



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 1: (a) Synthetic image with initial point s. (b) Detection of first keypoint k_1. (c) Detection of second keypoint k_2. (d) Fast March propagation from point s'. (e) Minimal path on the curve. (f) Minimal path in background region. (g) Curve and keypoints detected after 5 iterations. (h) Curve and keypoints detected after 8 iterations. (i) Final Image with ordered keypoints and terminating point marked by 'x'.

illustration. The source point set S is augmented by including the new keypoint k_1 and the set S now becomes $s \cup k_1$. An associative map $m : S \mapsto S$ is generated that defines an *origin point* for every keypoint in S. For a new keypoint, the origin point is the point closest in the set S to the keypoint. Hence, for k_1, the origin point is the initial point s. Though the initial point s does not have an origin point explicitly because it is not a keypoint, we assign k_1 as the origin point of s. The reason is that the minimal path for two points is symmetric and can be found by propagating the Fast Marching algorithm from either point (in this case s or k_1). Therefore, if we started with k_1 as the initial point in source set S, s would be the keypoint determined by the procedure $FMM(S, \lambda)$. Hence

$$m(s) = k_1, \tag{7}$$

$$m(k_1) = s. \tag{8}$$

Now, the Fast Marching propagation is carried out from the updated set S and the next keypoint k_2 is detected as shown in Figure 1c. Using the minimal path back propagation algorithm $MinimalPathbk(S, k_2)$, the point s^* in S that is a part of the minimal path from k_2 to S is identified. s^* is the origin point of k_2 and $m(k_2)$ equals s^*. In our example, origin point s^* is the initial point s. Next, we find the origin point s' of the point s^* from the map m (s' is the point k_1 for our example). If there exists a continuous path of desired feature points with low potential values between the points s' and k_2, then the minimal path between s' and k_2 should also pass through the vicinity of s^*. Figure 1e shows the minimal path from s' and k_2 in the case where the minimal path overlaps with the curve. When there is no portion of a curve on this minimal path, the path does not pass through the neighborhood of s^* as illustrated in Figure 1f. We denote the Euclidean distance from a source point set s' to a point k_2 as $L(s', k_2)$. The Fast Marching propagation required to determine $L(s', k_2)$ is denoted by $FMM(s', k_2)$. Figure 1d depicts this Fast March propagation. From Figure 1e, we conclude that if the minimal path between s' and k_2 lies on the curve, then

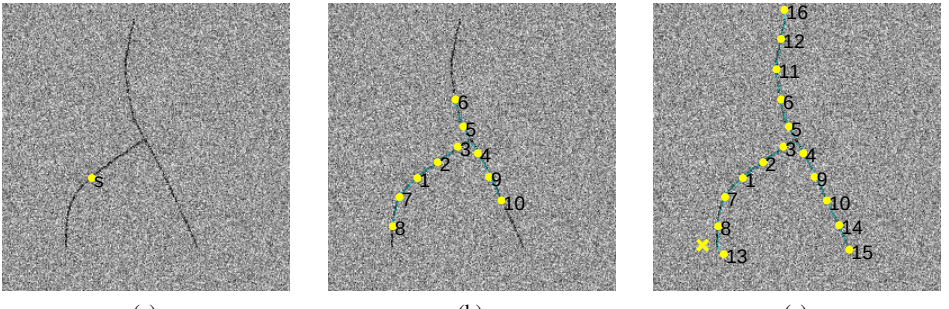
$$L(s', k_2) \approx L(s', s^*) + L(s^*, k_2). \tag{9}$$

As all keypoints are detected sequentially with a fixed Euclidean distance parameter λ, the Euclidean distance L between neighboring keypoints is close to λ. The point s^* is the closest point to the keypoint k_2 in the set S. Hence, by definition of the procedure $FMM(S, \lambda)$, the distance between s^* and k_2 is close to λ. Similarly, The point s' should be at a distance of λ from s^* because s' is the origin point of s^*. This fact is clear from Figure 1e. According to Equation 9, the Euclidean distance $L(S', k_2)$ should then be approximately 2λ if the minimal path from S' to k_2 passes through s^*. If Equation 9 does not hold (like in Figure 1f), then the curve detection procedure can be terminated and the algorithm outputs that there is no curve detected. Otherwise, this procedure of finding keypoints is recursively carried out and the subsequent keypoints are identified as k_2 in algorithm *CurveDetection* that is given below. The point s^* determined at each iteration is either the initial source point s or a keypoint. A *tolerance error* value ε is specified for the termination. We used an ε value of 0.2λ or 10% of the total length 2λ for the algorithm. The intermediate results of the algorithm are shown in Figure 1g and 1h. The complete curve with keypoints (including the initial point) is shown in Figure 1i. The keypoints are labeled according to the order of their appearance. The last keypoint for which Equation 9 does not hold and the algorithm terminates is called the *terminating point* (represented by 'x' in Figure 1i). For a simple curve with no branches, the estimated length of the detected curve should lie within λ distance of the actual length.

The above algorithm can also be used on more complex curves . Figure 2(a) illustrates a

Algorithm *CurveDetection*
Input: Image Im, potential function Φ and initial source point set S.
Output: Detected Curve C

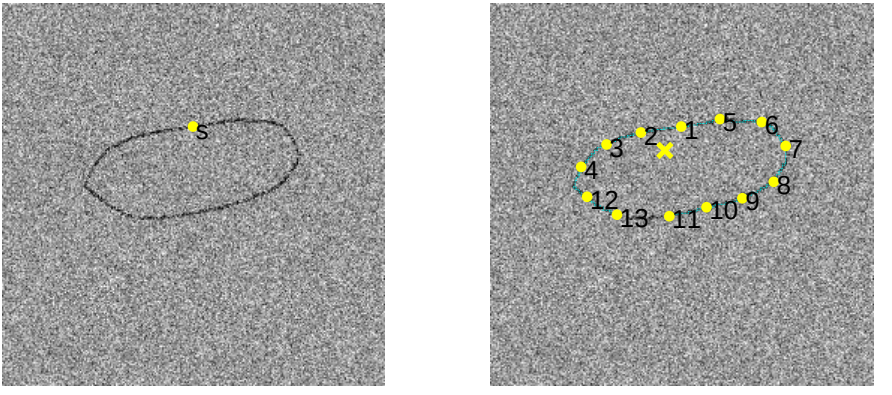
1. Start with initial source point set S containing an arbitrary point s on the curve.
2. Set $StopDetection = FALSE$.
3. Use $FMM(S,\lambda)$ to find keypoint k_1.
4. Run $MinimalPathbk(S,k_1)$ and initialize curve C to contain the minimal path between S and k_1.
5. Set $S \leftarrow S \cup k_1$.
6. Set $m(k_1) = s$ and $m(s) = k_1$.
7. **while** $StopDetection = FALSE$
8. **do** Run $FMM(S,\lambda)$ to find new keypoint k_2.
9. Run $MinimalPathbk(S,k_2)$ and find the origin point s^* in set S.
10. Set curve C' to contain the minimal path between S and k_2.
11. Compute point $s' = m(s^*)$.
12. Use $FMM(s',k_2)$ to calculate $L(s',k_2)$.
13. **if** $|L(s',k_2) - 2\lambda| < \varepsilon$
14. **then** $C \leftarrow C \cup C', S \leftarrow S \cup k_2$ and $m(k_2) = s^*$.
15. **else** $StopDetection = TRUE$.
16. **repeat**



(a) (b) (c)

Figure 2: (a) Curve with branches and source point s. (b) Intermediate result from the algorithm after 9 iterations. (c) Final curve detection with keypoints and final terminating point labeled by 'x'.

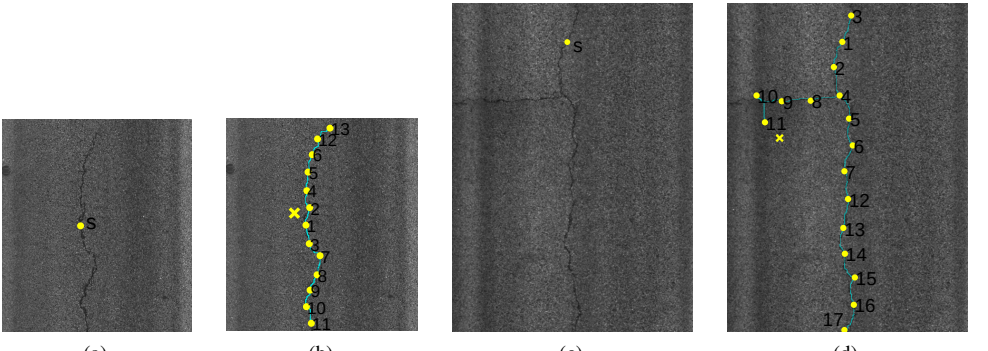
synthetic image that has a curve with branches embedded in a random background of higher mean intensity. An arbitrary source point s on the curve is provided as input to the algorithm. Figure 2(b) demonstrates the intermediate result of the algorithm on the image in which the curve and the keypoints detected are labeled. Figure 2(c) shows the final result with the complete curve, all ordered keypoints and the terminating point. Figure 3(a) is a synthetic image with a closed curve embedded in a random background and Figure 3(b) labels the detected curve with keypoints. A more generalized version of the described algorithm is being investigated to deal with the detection of complex curves with both open and closed curve sections.



(a) (b)

Figure 3: (a) Closed curve with initial point s. (b) Final curve detection with keypoints (including final terminating point labeled by 'x'.

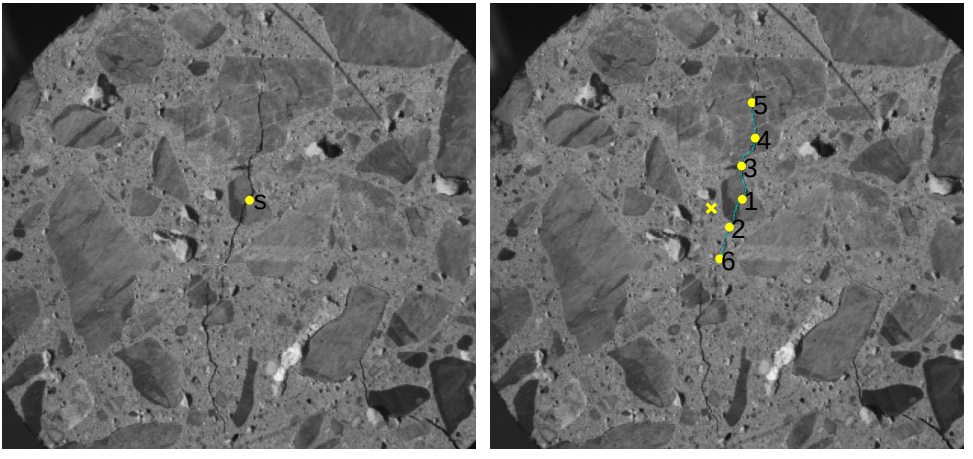
4 Experimental Results



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

Figure 4: (a) Asphalt crack image with initial point s. (b) Final crack detection with keypoints. (c) Crack image with multiple branches. (d) Final crack detection with keypoints.

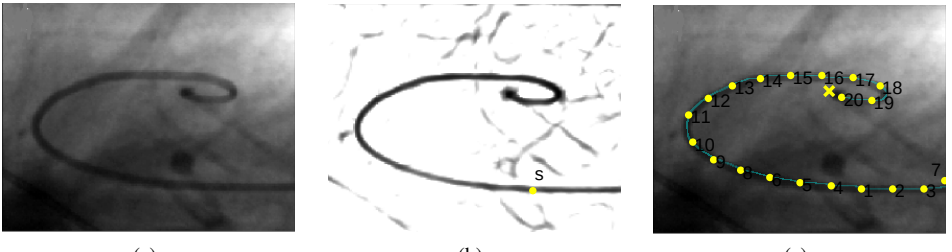
Our algorithm is applied to detect cracks in civil structures and detect thin curve like features in medical images. Cracks in structures can be modeled as curves that are darker in the intensity than the background. Recent studies [13, 15] show that automatic detection of cracks in these structures is very challenging because of multiple textures, shadows, variable lighting, irregular background and high noise present in the images. Most algorithms use multiple heuristic parameters that help to detect cracks for only a small data set of images. Hence, it is useful to have an algorithm that can detect cracks in local regions using only one user defined crack point. Figure 4(a) shows an asphalt pavement image with a user supplied point on the crack. We use our algorithm and detect the crack with high accuracy. Figure 4(b) shows the detected crack, ordered keypoints and the terminating point. Figure 4(c), 4(d) and attached video in supplementary material illustrate crack detection for a complex asphalt crack with multiple branches. We also used a concrete image with a highly variable back-



(a) (b)

Figure 5: (a) Concrete image with initial point *s* (b) Final crack detection with keypoints

ground for the experiments. Figure 5(a) shows the concrete structure image with a starting point *s* on the crack. Our algorithm detects the crack accurately in the local neighborhood of the user supplied point. The final result is given by Figure 5(b). Our algorithm can be used to detect features appearing in medical applications such as bone cracks, thin capillaries and optical nerves. We tested our algorithm on a medical image of a catheter tube shown in Figure 6(a). A potential function based on the Laplacian is used to provide contrast between the desired feature and the background. Figure 6(b) illustrates the potential function image. Figure 6(c) shows the detected curve, keypoints and terminating point for the medical image.



(a) (b) (c)

Figure 6: (a) Catheter tube image. (b) Edge based potential image. (c) Final curve detection with keypoints

For quantitative evaluation of our results, we calculated the False Positive (FP), False Negative (FN) and True Positive (TP) rates for the images. For real images that have features like cracks, ground truth accuracy itself is suspect. To overcome this problem, Kaul et al. [11] devised the scaled buffered distance (SBD) metric, which is a modification of the Hausdorff distance. BD measures the distance between two curves and the values lie between 0 and 1. The values 0 and 1 indicate the best and worst performance respectively. The results on all displayed figures are shown in Table 1. The algorithm results on the synthetic images are very good because the background does not have features similar to the detected

Table 1: Validation of Synthetic and Real Images

Rate	Image Number						
	Figure 1	**Figure 2**	**Figure 3**	**Figure 4(a)**	**Figure 4(c)**	**Figure 5**	**Figure 6**
TP%	93.8	96.1	90.9	99.1	97.2	97.8	88.1
FP%	0	0	0	16.3	18.1	4.3	12.5
FN%	6.2	3.9	9.1	1.3	10.1	8.1	4.6
BD	0.07	0.05	0.09	0.19	0.26	0.13	0.11

curve. Only small FN rates are encountered because of the termination criterion given by the algorithm. The ground truth for crack data was collected from structural experts. They were asked to represent the crack with zero-width curves in the local neighborhood of the initial source point s. For crack images in asphalt structure and the medical image, FP rates are moderately high because the background texture has uniformly dark regions that have properties similar to cracks. FN rates in crack images arise because there are significant breaks in crack locations and the crack features cannot be connected by continuous curve. SBD values for our algorithm are good indicating that the ground truth curve and the detected curve overlap with each other. The results of the algorithm on synthetic and real data demonstrate the validity of our algorithm. We also conducted a preliminary investigation on the influence of the Euclidean length interval λ (interval between successive keypoints) and the *Tolerance Error* value. Features like cracks sometimes do not form a continuous path and have breaks between them. If the user wants to connect these cracks together, a higher value of λ should be provided. A lower value of λ will lead to faster termination. When the potential function does not provide enough contrast between the features and the background even in a local neighborhood, higher *Tolerance Error* value needs to be used. The effect of parameters λ and *Tolerance Error* will be investigated more in the future.

5 Conclusions and Future Work

We have presented a novel algorithm for detecting curves with unknown endpoints based on minimal path techniques. We showed that this algorithm can detect curves using any arbitrary point on the desired curve as the sole user input. Synthetic and real image data were used to validate our algorithm. Specifically, the algorithm was applied to detect cracks in variable background structural images and to detect thin elongated medical features. In the future, more features like optical nerves, thin capillaries and bone cracks can utilize this algorithm. The algorithm can also be generalized to detect closed curves and curves with complex topologies. The algorithm is also easy and straight-forward to extend to volumetric data sets for the extraction of 3D (and even higher dimensional) curves.

6 Acknowledgement

The authors wish to thank Prof. Laurent Cohen at the University of Dauphine, France for some helpful conversations and for sharing the raw image used in Figure 6. This paper was sponsored by NSF Grant CCF-0728911.

References

[1] R. Ardon, L. D. Cohen, and A. Yezzi. Fast surface segmentation guided by user input using implicit extension of minimal paths. *Journal of Mathematical Imaging and Vision*, 25(3):289–305, 2006.

[2] F. Benmansour and L. D. Cohen. Fast object segmentation by growing minimal paths from a single point on 2D or 3D images. *Journal of Mathematical Imaging and Vision*, 33(2):209–221, 2009.

[3] S. Bonneau, M. Dahan, and L. D. Cohen. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *IEEE Transactions on Image Processing*, 14(9):1384–95, 2005.

[4] L. D. Cohen. Multiple contour finding and perceptual grouping using minimal paths. *Journal of Mathematical Imaging and Vision*, 14(3):225–236, 2001.

[5] L. D. Cohen and R. Kimmel. Global minimum for active contour models: a minimal path approach. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 666–673, San Francisco, CA, USA, 1996. IEEE.

[6] L. D. Cohen and R. Kimmel. Global minimum for active contour models: a minimal path approach. *International Journal of Computer Vision*, 24(1):57–78, 1997.

[7] T. Deschamps, J. M. Letang, B. Verdonck, and L. D. Cohen. Automatic construction of minimal paths in 3D images: an application to virtual endoscopy. In *Computer Assisted Radiology and Surgery. Proceedings of the 13th International Congress and Exhibition*, pages 151–5, Amsterdam, Netherlands, 1999. Elsevier Science.

[8] S. Guo and B. Fei. A minimal path searching approach for active shape model (ASM)-based segmentation of the lung. *Proceedings of the SPIE - The International Society for Optical Engineering*, 7259:72594B1–8, 2009.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321 – 31, 1987. ISSN 0920-5691.

[10] V. Kaul, Y. Tsai, and R. Mersereau. A quantitative performance evaluation of pavement distress segmentation algorithms. In *Transportation Research Board Meeting*, Washington D.C, USA. Brief summary supplied as supplemental material buffereddistance.pdf, 2010.

[11] K. Li and B. Fei. A new 3D model-based minimal path segmentation method for kidney MR images. In *2nd International Conference on Bioinformatics and Biomedical Engineering*, pages 2342–2344, Shanghai, China, 2008.

[12] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999.

[13] Y. Tsai, V. Kaul, and R. Mersereau. Critical assessment of pavement distress segmentation methods. *ASCE Journal of Transportation Engineering*, 136:11–19, January 2010.

[14] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

[15] K. C. P. Wang. Challenges and feasibility for comprehensive automated survey of pavement conditions. In *Proceedings of the International Conference on Applications of Advanced Technologies in Transportation Engineering*, pages 531–536, 2004.

[16] P. Yan and A. A. Kassim. Medical image segmentation using minimal path deformable models with implicit shape priors. *IEEE Transactions on Information Technology in Biomedicine*, 10(4):677–84, 2006.