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

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 objects in medical images. This procedure can also be extended to closed curves and more complex topologies consisting of both closed curves and open curves. Minimal path technique is based on the computation of the *geodesic distance map* $U(x)$ that seeks to minimize weighted distance between two points p_1 and x.

$$U(x) = \min_{\gamma \in \mathscr{A}_{p_1,x}} \int_{\gamma} \Phi(\gamma(s))\mathrm{d}s. \tag{1}$$

where Φ is the potential that takes lower values at the desired feature points, $\mathscr{A}_{p_1,x}$ is the set of all paths connecting p_1 to x, and γ is a curve between p_1 and x. For example, in the case of a pavement crack image, the potential ϕ can be chosen to be based on intensity values because cracks are darker than their neighboring surroundings. The curve γ that minimizes the weighted distance given in Equation 1 is called the *minimal path* or the *geodesic curve* and is denoted by $C_{p_1,x}$. In addition, another quantity called the *Euclidean distance map* $L(x)$ is computed along the geodesic curve $C_{p_1,x}$. It is given by

$$L(x) = \int_{C_{p_1,x}} \mathrm{d}s. \tag{2}$$

L gives the Euclidean distance of the minimal path $C_{p_1,x}$ between p_1 and x. The computation of the geodesic distance map U is equivalent to solving a non-linear partial differential equation called the Eikonal's equation given by

$$||\nabla U(x)|| = \Phi(x) \tag{3}$$

This partial differential equation is solved using an upwind scheme called the *Fast Marching method* and the distance map $U(x)$ is calculated for all grid points x until the endpoint of interest is reached p_2. 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 reduce the user interaction, we propose a novel algorithm to detect curves using one arbitrary point on the curve. It does not require knowledge of the two endpoints. This algorithm extends the Minimal Path for KeyPoint Detection (MWKPD) approach developed by Benhamasour and Cohen [1]. The MWKPD approach 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 source point set S (S contains an endpoint 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 (not necessarily an endpoint) on the curve. We briefly discuss the algorithm next.

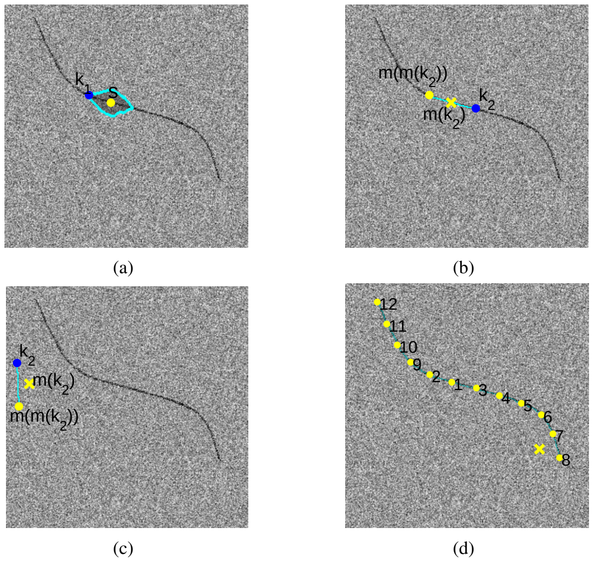

Figure 1: (a) Synthetic image with initial point s and first keypoint k_1. (b) Minimal path on the curve. (c) Minimal path in background region. (d) Final Image with ordered keypoints, detected curve and terminating point marked by 'x'.

The algorithm starts by identifying the first keypoint k_1 after propagating the Fast Marching boundary from a source point set S, which contains an arbitrary point s on the curve . Figure 1(a) shows an arbitrary initial point s and the first keypoint k_1. We also keep track of the nearest neighbor of each point in S using a map $m : S \mapsto S$. For example, the nearest neighbor for k_1 is s and vice versa. The next keypoint k_2 is computed after adding point k_1 to the source point set S and propagating the Fast Marching boundary from the updated source point set S. In our algorithm, the relative location of points k_2, $m(k_2)$ (nearest neighbor of k_2) and $m(m(k_2))$ (nearest neighbor of $m(k_2)$) determines whether these three points lie on the curve of interest or on the background region. If there exists a continuous path of desired feature points (modeled by a curve) with low potential values between k_2 and $m(m(k_2))$, then the minimal path between k_2 and $m(m(k_2))$ should pass through the vicinity of the $m(k_2)$ as shown in Figure 1(b) and satisfy the condition

$$L(m(m(k_2)),k_2) \approx L(m(m(k_2)),m(k_2)) + L(m(k_2),k_2). \tag{4}$$

If no portion of the desired curve of interest lies on the minimal path between k_2 and $m(m(k_2))$, Equation 4 does not hold as shown in Figure 1(c). As long as the continuous path condition given by Equation 4 is met, new keypoints are recursively computed by the algorithm, and the algorithm terminates when the continuous path condition fails to hold. Figure 1(d) shows the ordered keypoints, the complete detected curve and terminating point (marked by 'x'). The paper provides more extensive experimental results on both synthetic and real images and demonstrates that our algorithm can be applied effectively to detect objects or features, which can be modeled as curves.

[1] 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.