DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-378

Low-Level Track Finding and Completion using Random Fields

Tu-Thach Quach, Rebecca Malinas, Mark W. Koch; Sandia National Laboratories; Albuquerque, NM, USA

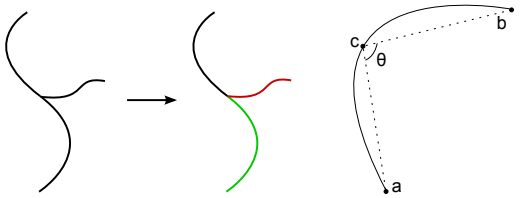
Abstract

Coherent change detection (CCD) images, which are products of combining two synthetic aperture radar (SAR) images taken at different times of the same scene, can reveal subtle surface changes such as those made by tire tracks. These images, however, have low texture and are noisy, making it difficult to automate track finding. Existing techniques either require user cues and can only trace a single track or make use of templates that are difficult to generalize to different types of tracks, such as those made by motorcycles, or vehicles sizes. This paper presents an approach to automatically identify vehicle tracks in CCD images. We identify high-quality track segments and leverage the constrained Delaunay triangulation (CDT) to find completion track segments. We then impose global continuity and track smoothness using a binary random field on the resulting CDT graph to determine edges that belong to real tracks. Experimental results show that our algorithm outperforms existing state-of-the-art techniques in both accuracy and speed.

Introduction

Multiple synthetic aperture radar images taken at different times of the same scene can be combined to produce coherent change detection images that can reveal subtle surface changes such as those made by tire tracks [1, 2]. The CCD images, however, are noisy due to SAR speckle, vegetation, shadows, and other weather related phenomena. These undesirable noise sources make it difficult to automatically identify and label vehicle tracks. In addition, SAR images are unimodal [3], making it difficult to identify high-quality image features that can differentiate tracks from background noise.

The difficult nature of this problem is exemplified by the limited success of existing techniques [4, 5]. Given a search cue, e.g., starting location of a path, the technique in [4] finds the vehicle track by tracing parallel lines (tire tracks). The proposed method can find a single track provided that the user supplied the initial search cue. A related method uses cubic spline fitting to extract vehicle tracks [5]. It is limited to a single non-overlapping track. In practice, a scene can have an arbitrary number of tracks, including no tracks. In addition, these tracks may come from different types of vehicles. Any automatic technique must be able to account for these conditions. A recent method addresses some of these limitations by finding the simplest set of tracks that explains the observed data [6]. It is a greedy method that finds one track at a time, until the objective function can no longer decrease. The obtained tracks, which are line segments, are iteratively merged to form full tracks. The merging step is important as the initially obtained tracks may not be complete and the merging procedure allows the algorithm to discover missing parts of tracks. It is, however, also computationally expensive as it considers every possible pairwise merges recursively until convergence. Furthermore, the approach uses a parallel track template to find candidate tracks. It is unclear how that approach can be generalized to find single tracks, such as those made by motorcycles, or parallel tracks made by various vehicle sizes.



(a) (b)

Figure 1: Branching (a): the original track (left) is split into three tracks (right) at the intersection. Linearizing (b): a track is split until the angle θ is larger than a threshold. Best viewed in color.

This work presents an approach that can find different types of tracks. We first identify a track feature based on the Hessian of the image surface that can be used to initially identify good track segments. We then find the remaining segments using the CDT where the initially identified segments form the constrained edges. We further reduce the set of edges by leveraging the fact that tracks tend to be smooth. We determine which edges of the augmented CDT belong to real tracks by forming a binary random field on the edges. Our energy function consists of unary and submodular pairwise terms for which a globally optimal solution can be found in polynomial time using graph cut.

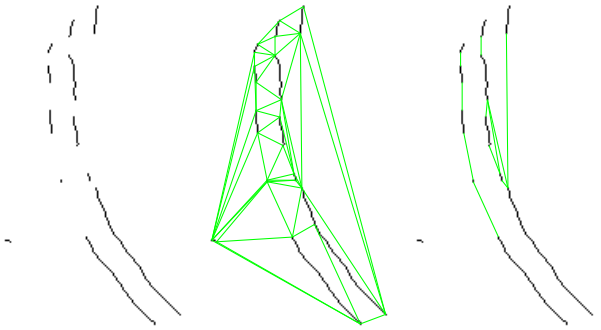
We are, of course, not the first to use the CDT to find contours. Our approach is similar to the one proposed in [7]. In that work, the CDT is used to find boundaries in natural images. They use the probability of boundary (*Pb*) feature [8] to find edges and form a binary random field on the CDT. In contrast, we are interested in finding tracks, which are not natural image boundaries. Furthermore, we augment the CDT by leveraging the fact that tracks tend to be smooth, resulting in a smaller graph. Our energy function is also different. Specifically, our energy function allows for a global solution via graph cut. In contrast, the original work uses belief propagation to find an approximate solution to a higher-order energy function.

The next section describes the track feature and the construction of the augmented CDT. We then present the submodular energy function that allows for a global solution. Following that, we present experimental results demonstrating the effectiveness of our algorithm on real and synthetic data sets. Concluding thoughts are provided in the last section.

Track Completion

Tracks can be viewed as ridges (or trenches) on the image surface. Given input image I, our first step is to compute a ridge feature for each pixel [9]. Specifically, the ridge feature at pixel i is λ_i, the largest, in magnitude, eigenvalue of the Hessian matrix

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-378



(a) (b) (c)

Figure 2: Example CDT output and removal of irrelevant completion edges: (a) tracks found by thresholding process, (b) CDT output showing completion edges in green, (c) relevant completion edges. Completion edges bridge many missing parts of the vehicle track. The removal step keeps meaningful completion edges.

of I at pixel i. We generally filter I with a Gaussian kernel so that the Hessians are more stable. Next, we form binary image I_{bw} by thresholding the ridge features: $I_{bw}(i) = 1$ if $\lambda_i > \tau$. The initial set of tracks is formed by thinning I_{bw}, branching, and recursively splitting each track until they are approximately linear. Figure 1 illustrates the branching procedure, which splits a track into multiple tracks at each intersection, and the linearizing process, which recursively splits a track until the minimum angle formed by any interior point (c) along the track and its two end points (a and b) is above a threshold.

The resulting set of tracks forms the *constrained* edges of the CDT. The new edges formed by the CDT are the *completion* edges. These completion edges bridge the gaps between constrained edges. Figure 2(b) shows an example CDT output. The black lines are constrained edges and the green lines correspond to completion edges. The completion edges discover the missing parts of the vehicle track. The example also shows that many completion edges are not meaningful, especially those that run in the middle of the parallel tracks. Using prior knowledge that tracks tend to be smooth, we keep only relevant completion edges as follows. We keep a completion edge if the angle formed by the completion edge and one of its neighboring constrained edges is larger than a threshold. This condition must hold at both ends of the completion edge. In other words, we want the completion edges to be relatively straight with respect to its neighboring constrained edges. The result of this removal procedure is illustrated in Figure 2(c).

Energy Minimization

In order to determine which edges belong to real tracks, we impose a binary random field on the obtained CDT graph. Let the set of edges of the CDT graph be E, the set of constrained edges be $E_C \subset E$, and $N \subset E \times E$ be the neighborhood system on E that stores adjacent edges. For each edge $i \in E$, we assign a label $x_i \in \{0,1\}$, where $x_i = 1$ indicates that edge i is part of a track. The optimal labeling minimizes the following objective function:

$$\sum_{i \in E} f_i(x_i) + \sum_{i,j \in N} [x_i \neq x_j][i \in E_C \vee j \in E_C] g(i,j), \tag{1}$$

where $f_i(x_i)$ is the cost of assigning label x_i to edge i, $g(i,j)$ is a smoothing cost based on the angle formed by edges i and j, and $[\phi]$ is an indicator function on predicate ϕ.

Let the set of pixels corresponding to edge i be $\mathscr{P}(i)$ and $\lambda^i = \frac{1}{|\mathscr{P}(i)|} \sum_{j \in \mathscr{P}(i)} \lambda_j$ is the average ridge feature of the pixels along edge i. Our label cost is

$$f_i(x_i) = -\log p(\lambda^i | x_i). \tag{2}$$

Let μ_0 and σ_0 be the mean and standard deviation of the background ridge feature, respectively, and μ_1 and σ_1 be the mean and standard deviation of the track ridge feature, respectively. By the central limit theorem, we expect λ^i to be normally distributed given x_i:

$$p(\lambda^i | x_i = 0) \approx \mathscr{N}(\mu_0, \sigma_0 / \sqrt{|\mathscr{P}(i)|}) \tag{3}$$

and

$$p(\lambda^i | x_i = 1) \approx \mathscr{N}(\mu_1, \sigma_1 / \sqrt{|\mathscr{P}(i)|}). \tag{4}$$

We note that these equalities only hold approximately as adjacent pixels are not strictly independent. The means, μ_0 and μ_1, and standard deviations, σ_0 and σ_1, can be obtained from ground-truth images. Note that our objective function does not explicitly model track lengths, but they are captured in $f_i(x_i)$. The longer an edge is, the more confident we are about its label.

Let the angle formed by edges i and j be θ_{ij}. The smoothing cost is a logistic function

$$g(i,j) = \frac{\alpha}{1 + \exp(-\beta(\theta_{ij} - \theta_0))}, \tag{5}$$

where α, β and θ_0 are constants. Note that the smoothing cost only applies when the adjacent edges have different labels and at least one of them must be a constrained edge; we do not enforce smoothness on adjacent completion edges as their purpose is to fill in the gaps between constrained edges. Once again, the smoothing cost favors large angles as these incur larger penalties.

Since our objective function is submodular, a globally optimal solution can be obtained via graph cut [10]. As a final step, we remove any short, isolated edges.

Experimental Results

We demonstrate the effectiveness of our approach using real SAR CCD images from a publicly available data set [11]. The results are shown in Figure 3. The first column shows the input CCD images. The middle column shows the images obtained from thresholding the ridge features. The last column shows the final results obtained by our algorithm. As described earlier, the thresholded images contain disconnected pieces of tracks. By forming the CDT using these pieces as constrained edges, our algorithm fills in the missing pieces, resulting in smoother, more contiguous tracks.

In order to quantitatively compare the performance of our algorithm, we use the image set tested by the greedy track finder [6]. This image set consists of two subsets, *dark* and *medium*, representing varying track thicknesses. Each subset consists of 40 CCD images of size 800×600 containing simulated tire tracks of various curvatures. Each test image is generated from a real SAR

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-378

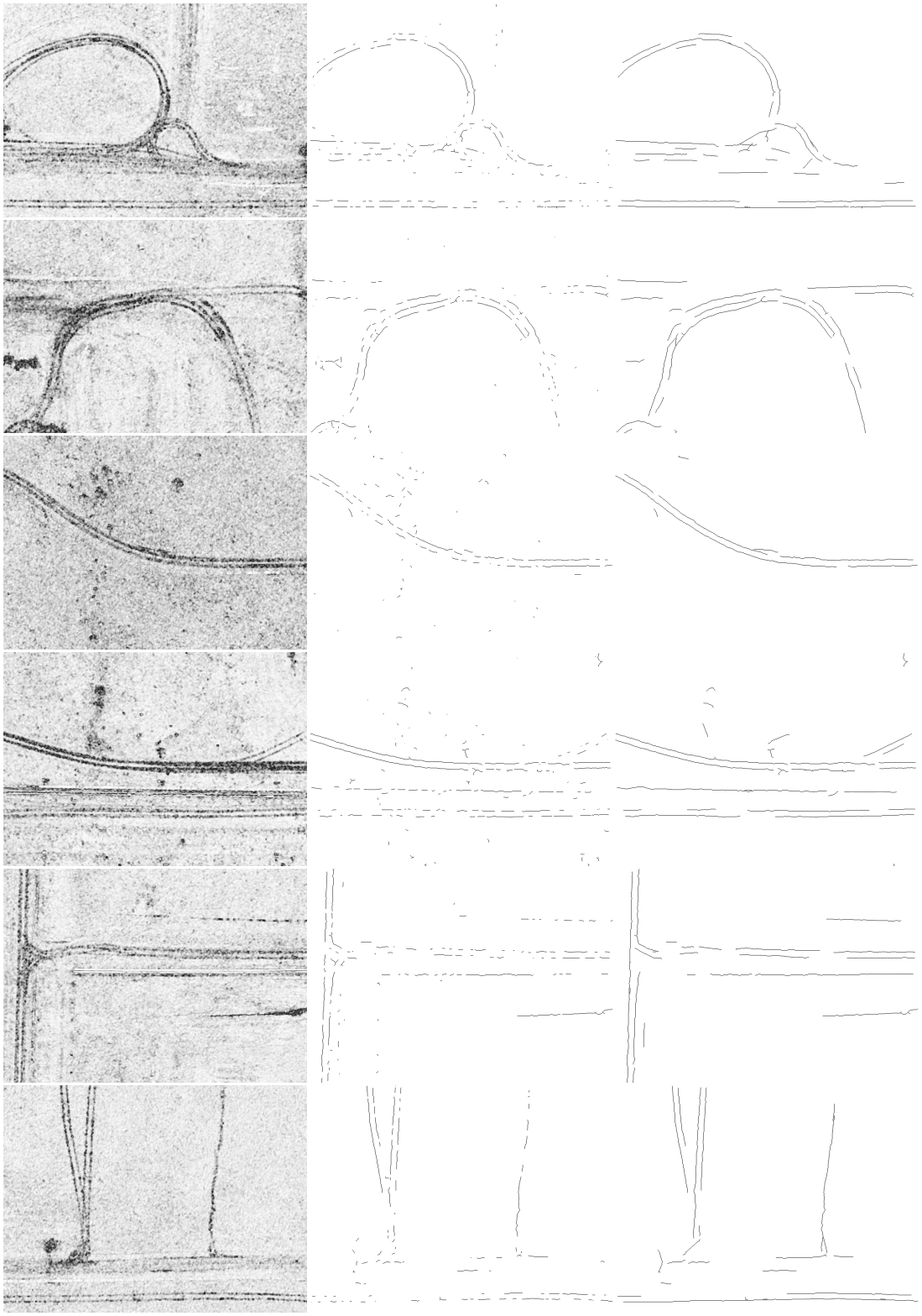
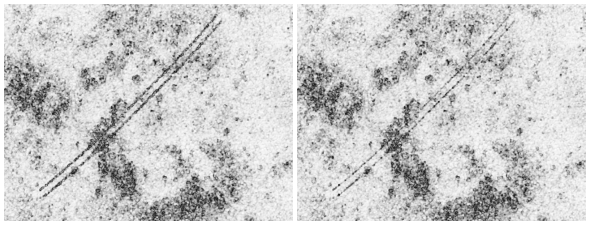


Figure 3: Example results on real SAR CCD images: input CCD images (left), thresholded images (middle), and output images (right). The thresholded images contain broken pieces of tracks and the algorithm fills in the missing pieces resulting in smoother, more contiguous tracks.

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-378



(a) (b)

Figure 4: Example test images: (a) dark, (b) medium. Each set consists of 40 images of size 800×600 containing various track curvatures and background clutters. The average track length is 705.18 pixels.

image pair (from the same publicly available data set) and contains a single randomly-generated tire track. Given a track and a pair of SAR images of the same scene, the track is added to the output CCD image by adding random (Gaussian) phase shifts to pixels along the track trajectory in the non-reference image of the SAR image pair [12]. The average track length is 705.18 pixels. Example images are shown in Figure 4. We note that this image set is not complete as it does not capture other aspects that are present in practice such as scenes containing multiple tracks made by different vehicle sizes.

We use the accuracy metric proposed in [6] to evaluate the performance of our algorithm:

$$\frac{TP}{GT + FA}, \tag{6}$$

where TP is the number of correctly detected track pixels, GT is the total number of track pixels, and FA is the number of non-track pixels incorrectly identified as track pixels. This accuracy metric is a real number between 0 and 1. It is 1 if and only if all track pixels are detected and there are no false alarms. As the number of false alarm pixels increases, accuracy decreases. Likewise, as the number of correctly detected pixels decreases, accuracy also decreases.

The tire track midline is taken to be the ground-truth track pixels. A track pixel is correctly detected if and only if there is a candidate pixel (classified by the algorithm as track) within a Euclidean distance of $r+5$ from it, where r is the tire track radius from the midline. The constant 5 is a buffer to accommodate for candidate track pixels that lie on or near the tire tracks. Any candidate pixel that does not have a corresponding ground-truth pixel is considered a false detection (FA).

We note that our algorithm actually finds the tire tracks, not the midlines. However, in order to compare our algorithm against the greedy algorithm, which only finds the midlines, we have to use the same ground-truth midlines. The mean accuracies for the two image sets are shown in Table 1. We use the same algorithm parameters to detect tracks in both image sets. The results show that the algorithm performs well on both sets. In particular, the accuracy for the medium data set increases significantly, from 0.84 to more than 0.98.

Unlike the greedy algorithm, which recursively finds and merges tracks until convergence, our algorithm finds the optimal set of tracks by solving the energy function once. In practice, this leads to significant computational advantage. On average, it takes less than one minute to find tracks in 4000-by-4000 images. In contrast, the greedy algorithm takes 30 minutes.

Table 1: Mean accuracies of the greedy algorithm [6] and the current algorithm on the *dark* and *medium* image sets.

	Greedy [6]	Current
dark	0.9767	**0.9941**
medium	0.8429	**0.9822**

Conclusion

The ability to find tracks is an important tool in the areas of security and surveillance. While CCD images offer the possibility of seeing subtle surface changes caused by vehicle tire tracks, developing automatic algorithms to find these tracks have been limited. This work presented an approach that can accomplish this task. Our approach exploits the fact that it is easier to find small pieces of tracks than whole tracks. Using these initial pieces, we can then discover the missing pieces to complete the tracks using the CDT. The best set of tracks is found via graph cut on a binary energy function that has a global solution. The resulting algorithm is computationally efficient and produces results that outperform the existing greedy algorithm.

Our current formulation does not take advantage of the fact that vehicle tracks may be parallel. Expanding our approach to account for this condition may further improve our algorithm. We defer investigating this problem to our future work.

Acknowledgment

This work was supported by PANTHER, a Laboratory Directed Research and Development (LDRD) Project at Sandia National Laboratories. For additional information about PANTHER, please contact Kristina Czuchlewski, Ph.D., krczuch@sandia.gov.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] Douglas G. Corr and Alex Rodrigues. Coherent change detection of vehicle movements. In Geoscience and Remote Sensing Symposium, pages 2451–2453. IEEE (1998).

[2] Charles V. Jakowatz, Daniel E. Wahl, Paul H. Wahl, Dennis C. Ghiglia, and Paul A. Thompson. Spotlight-mode synthetic aperture radar: a signal processing approach. Kluwer Academic Publishers Norwell, MA (1995).

[3] Jong-Sen Lee and Igor Jurkevich. Segmentation of SAR images. IEEE Transactions on Geoscience and Remote Sensing, 27 (1989), 6 pages 674.

[4] Miriam Cha and Rhonda Phillips. Automatic track tracing in SAR CCD images using search cues. In Signals, Systems and Computers (ASILOMAR), pages 1825–1829. IEEE (2012).

[5] Miriam Cha, Rhonda Phillips, and Michael Yee. Finding curves in SAR CCD images. In ICASSP, pages 2024–2027. IEEE (2011).

[6] Tu-Thach Quach, Rebecca Malinas, and Mark W. Koch. A model-based approach to finding tracks in SAR CCD images. In Computer Vision and Pattern Recognition Workshops. IEEE (2015).

[7] Xiaofeng Ren, Charles C. Fowlkes, and Jitendra Malik. Scale-invariant contour completion using conditional random fields. In International Conference on Computer Vision, volume 2, pages 1214–1221. IEEE (2005).

[8] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learn-

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-378

ing to detect natural image boundaries using brightness and texture. In Advances in Neural Information Processing Systems, volume 15 (2002).

[9] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. International Journal of Computer Vision, 30 (1998), 2 pages 117.

[10] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Machine Intell., 26 (2004), 2 pages 147.

[11] Wallace J. Bow Jr. Sandia SAR data collect 2006. Technical Report SAND2006-2290P, Sandia National Laboratories (2006).

[12] Eric Turner, Rhonda D. Phillips, Carol Chiang, and Miriam Cha. Inserting simulated tracks into SAR CCD imagery. In Autumn Simulation Multi-Conference. Society for Modeling & Simulation International (2012).