

Bayesian Tracking of Linear Structures in Aerial Images

Rui Gao and Walter F. Bischof
 Computing Science
 University of Alberta
 {rgao,wfb}@cs.ualberta.ca

Abstract

The interpretation of aerial images is difficult, especially for low-resolution images. Although solutions have been worked on for many years, performance of these systems is still not sufficient to be useful in practical applications. One potential solution is to create systems composed of many specialized modules. We introduce one such module, a Bayesian tracker for linear structures, such as pipelines and access roads. We show that the tracker can be used successfully to detect these structures in low-resolution images.

1. Introduction

Remote-sensed images are used in many GIS applications, including map production and updating, urban and regional planning, agricultural and forestry assessment, detection of mineral resources, and many others. Over the past decades, large efforts have been put into the design of computerized image interpretation systems that should make it possible to cope with the tremendous amounts of remote-sensed data. Unfortunately, these systems are still not accurate and reliable enough to be useful in practical applications. Several strategies have been pursued to overcome this problem. First, one can design of semi-automatic systems, where the computer acts as an assistant to the human operator, taking over very simple tasks, and returning control to the operator whenever a problem is encountered [14, 15]. Second, one can focus on systems composed of many modules, each specialized on one particular image interpretation task. This paper presents one such module: it is concerned with the detection of (oil or gas) pipelines and oil-well access roads in low-resolution aerial images.

In the past, there have been many attempts at detecting linear structures, such as roads in high-resolution aerial images (e.g., [10, 7, 12, 13]). These systems make a number of assumptions about the appearance of roads in the remote-sensed images: for example, roads are elongated structures,

road surfaces are usually homogenous, and there is adequate contrast between roads and adjacent areas. Many of these systems can deal with difficult image interpretation problems, including complex road topologies near crossings, bridges, ramps, etc.; occlusions by ground objects such as vehicles, shadows, trees, etc.; and inadequate contrast due to road texture, illumination conditions, weather conditions, and more. Many road tracking systems rely on line or edge tracking for road detection and focus on tracking the dynamic characteristics of roads, such as changes in orientation and texture. Some of the systems use a search approach to find the next pixel to link to the road network. For instance, the heuristic method proposed in [8] starts from pixels with a maximal second derivative and constructs lines by adding the appropriate neighbors to these lines, using distance and angle difference information. Bordes [3] used an extended Kalman filter to recursively predict the road axis, where feedback from matching the predicted road profile to the reference road profile is used to estimate the optimal tracking state. A similar, more general approach was used in [13] where roads are tracked using a particle filter. The filter approximates the optimal state by a particle set and a corresponding weight set. Starting with a set of particles, each with the same initial probability, the algorithm gradually adjusts the weights of each particle during the evolution process.

Several pattern-matching approaches to road tracking assume that road segments can be represented by specific patterns. For example, McKeown and Delinger [10] used patterns describing road segments that were selected by a human operator. The road direction and width could be estimated from these patterns. A road section corresponding to the next step in the direction was extracted and a cross-correlation between the predicted new road section and the tracking model was carried out. In [7], a region-based tracker used snakes to find road boundaries in high-resolution images. Regions and background were assumed to have constant intensity, a condition of collinearity was used to keep the two road borders parallel, and a smoothness constraint was used to stabilize local deformation.

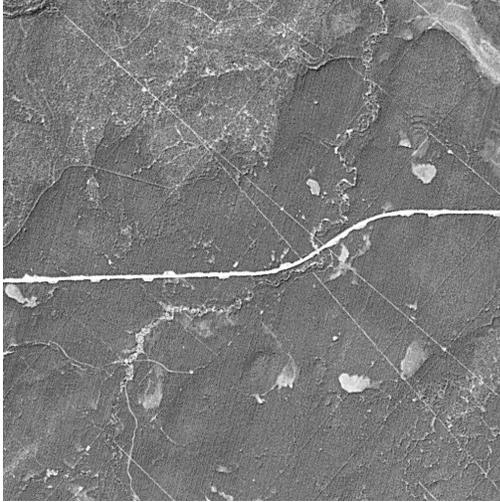


Figure 1. Example of an aerial image

The systems described above all require high-resolution aerial images that permit the extraction of road profiles and road edges. Unfortunately, high-resolution images are often not available for remote areas. In low-resolution images (with a resolution of at least 5 meters per pixel), pipelines and access roads may be only a single pixel wide, eliminating the possibility of extracting profiles orthogonal to the road direction. This is illustrated in Figure 1, which shows an aerial image of size 1000×1000 pixels with a resolution of 5 meters per pixel, containing a number of natural structures (hills, rivers, lakes) and several human-made structures. Clearly visible in the middle of the image is an S-shaped road. In addition, several long, straight, but faint lines are visible, indicating the presence of pipelines and associated access roads. Existing road tracking systems have no problem detecting the S-shaped road, but fail to detect the others. The work presented here is concerned with the detection of these linear structures.

2 Overview

The Bayesian line tracker described here is part of a larger system [5]. Several of the components, those concerned with the detection of linear segments, have been described elsewhere [6] and are summarized only briefly, namely the construction of a local orientation map (Section 2.1), the extraction of line segments using an orientation-weighted Hough transform (Section 2.2), and the combination of line segments into linear structures using a Markov random field (Section 2.3). In Section 3, we introduce the Bayesian tracking model and the tracking process. The tracker is responsible for tracking curved roads and for tracking the curved connections between linear segments of

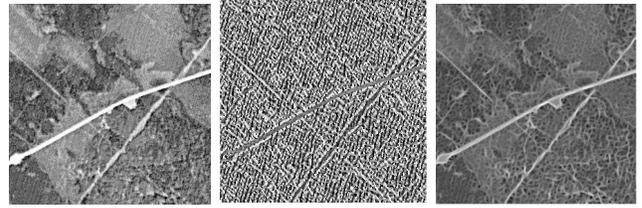


Figure 2. Left: Aerial image of size 300×300 pixels with resolution of 5 meters per pixel. Middle: Orientation map with 10 orientations denoted by different grey levels. Right: Texture map representing the magnitude of the Gabor filter response

linear structures. In Section 4, we present experimental results, and in Section 5, we present conclusions.

2.1 Local Orientation Analysis

To determine the dominant local orientation at all image positions, the input images are filtered with even Gabor filters

$$G(x, y) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \cos\frac{2\pi x'}{\lambda} \quad (1)$$

with

$$x' = x \cos \theta + y \sin \theta \quad (2)$$

$$y' = -x \sin \theta + y \cos \theta \quad (3)$$

where λ represents the wavelength of the cosine carrier, σ defines the scale of Gaussian envelope, and θ is the filter orientation. We use a bank of 10 Gabor filters with orientation θ uniformly distributed in the interval $[0, \pi)$, and we used $\lambda = 5$ and $\sigma = 2$.

The orientation map $o(x, y)$ is defined for each pixel (x, y) as the orientation θ of the Gabor filter with maximal response magnitude, and the texture map $g(x, y)$ is defined as the maximal response magnitude. The orientation map represents local image orientation whereas the texture map represents local texture [9]. Figure 2 shows an image of size 300×300 pixel and the corresponding orientation and texture maps. These two characteristics are used in the next step to extract linear segments in the images.

2.2 Orientation-Weighted Hough Transform

The Hough transform is used to detect lines by transforming image space into Hough space, selecting maxima in Hough space, and using these maxima to identify lines in image space. Each pixel in image space votes equally for

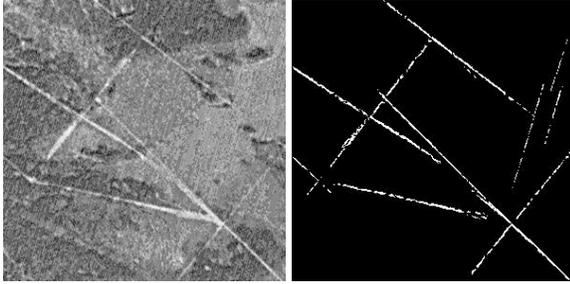


Figure 3. Line detection. Left: Input image of size 300×300 pixels. Right: Lines detected using an orientation-weighted Hough transform

all possible straight lines through the pixel. The detection of straight lines is thus susceptible to the presence of random noise and spurious line segments, which may generate false maxima in Hough space.

To reduce false contributions, we introduce a weighting scheme into the voting strategy. The orientation map (see Section 2.1) is used to assign weights in terms of how well the orientation $o(x, y)$ at pixel (x, y) matches the line orientation. Given the orientation map $o(x, y)$ (see Section 2.1), the contributing weight to accumulator cell (θ, ρ) in Hough space should be larger when θ is close to $o(x, y)$. The weight can thus be defined as

$$W_{\theta, o(x, y)} = |\cos(o(x, y) - \theta)|.$$

The orientation-weighted Hough transform is somewhat related to earlier work on Hough transforms for short lines and edge segments (e.g., [4]), but it generalizes that work, replacing heuristic grouping by a weighting scheme using Gabor filters. As Figure 3 indicates, this technique can detect straight lines while reducing the detection of spurious line segments. In the next Section, we describe how to label these line candidates and combine them together to identify linear structures.

2.3 Combining Linear Structures

The method introduced in the previous Section is able to identify linear structures, but they may be broken into many line segments, as seen in Figure 3. To complete the identification process, these line segments must be combined into linear structures. As described in detail in [6], we define a graph of all line segments, and over this graph we define a Markov Random field (MRF). The node and clique potentials of the MRF are defined in a way that captures the important properties of lines (e.g. line segments are long, have no gaps, and have a consistent texture) and line conjunctions (e.g. conjunctions have low curvature, end points

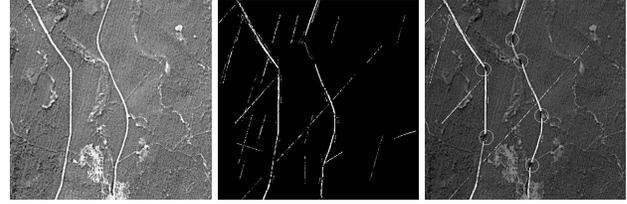


Figure 4. Line detection. Left: Input image of size 1000×1000 pixels. Middle: Lines detected using the orientation-weighted Hough transform. Right: Line labeling result on using Markov Random Field model. Spatially related line segments are connected by black lines.

are rare, intersections of lines are rare, etc.). The results of linear structure combination using the MRF is illustrated in Figure 4.

The MRF analysis achieves two goals. First, isolated line segments, which are unlikely to be part of an extended linear network, are eliminated. This is seen by comparing the results obtained with the Hough transform (Figure 4 middle) and the results obtained after applying the MRF model (Figure 4 right). Second, line segments that are likely to belong to the same linear structure are identified. This is seen by the black lines (in the white circles) connecting neighbouring line segments.

3 Bayesian Tracking Model

The methods described in the previous Section can successfully identify linear structures, and they can identify line segments that belong to the same structure. What is, however, still missing is a way to track those connections along a curved path (See Figure 4). This is the task of the Bayesian tracker presented in this Section.

The Bayesian tracking process involves the construction of a posterior probability density function of the current state based on the accumulated observations. First, we build a prediction model (state model) to find all possible continuations of a line, and we assign a probability to each predicted state. Second, we match the characteristics extracted at these predicted states to the reference characteristics using a matching model (measurement model). Third, we use the Bayesian model to recursively update the optimal tracking state. Finally, we discuss stopping strategies.

The main task of line tracking is to find the next appropriate line position based on past and current observations. A reference profile is estimated based on the geometry and spatial information of a line, namely direction and texture. In high-resolution aerial images (less than 5 meters

per pixel), the width of a road or pipeline can also be used. In low-resolution images, roads and pipelines are typically 1-2 pixels wide, and width is thus not a useful line characteristic.

The tracking process can be modeled as a time series by a state-space approach [1]. It involves two stages, prediction and matching. A prediction model is used to describe the state evolution, and the matching model is built up to relate the observation measurements to the reference profile. These two models are ideally suited for a Bayesian approach.

3.1 Prediction Model

The state vector at time k , denoted by X_k , contains all relevant information required to describe the linear structure tracking system. It can be defined as (see [12]):

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \quad (4)$$

where (x_k, y_k) is the line axis position, and θ_k is the line direction. According to [1], the state vector X_k depends on the previous state X_{k-1} and a process noise w_k ,

$$X_k = f_k(X_{k-1}, w_k).$$

In the line tracking system, the state X_k is updated by

$$X_k = \begin{bmatrix} x_{k-1} + d\cos(\theta_{k-1} + \theta'_k) \\ y_{k-1} + d\sin(\theta_{k-1} + \theta'_k) \\ \theta_{k-1} + \theta'_k \end{bmatrix}, \quad (5)$$

where θ' is the direction shift from state X_{k-1} to X_k , and d is a constant, representing the distance covered in each tracking step. The values of d and θ'_k are determined by the prediction scale. Generally, a larger prediction scale is more tolerant to occlusions but is more likely to generate incorrect predictions. Under the assumption that the upper bound of the curvature at time k is $\pi/6$, we let θ'_k be distributed uniformly in the interval $[-\frac{\pi}{6}, \frac{\pi}{6}]$, i.e. $\theta'_k \in [-\frac{\pi}{6}, -\frac{\pi}{6} + \varphi, \dots, \frac{\pi}{6}]$. The number of predicted states is determined by the value of φ : Smaller values make the prediction more sensitive to changes of the line curvature. In the prediction stage, we search for all the possible line states within a small neighborhood, as shown in Figure 5.

The state X_0 is initialized based on the results of line detector (see Section 2.2). The tracking direction θ_0 is set to the line orientation obtained from the line detector, (x_0, y_0) is set to the position of the starting pixel of the line, and the probability of $P(X_0)$ is set to 1. Then the state sequence is predicted recursively using Equation 5.

Each predicted state is assigned a probability $P(X_k|X_{k-1})$ called prediction probability. The line

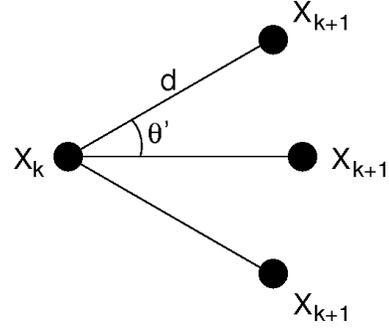


Figure 5. Find the possible state X_{k+1} based on previous state X_k within the interval distance d and direction change θ' .

curvature changes slowly within a small distance d , assuming that lines are fairly straight. The probability density of the prediction can be defined as

$$P(X_k|X_{k-1}) = \frac{1}{Z} |\cos(\theta_k - \theta_{k-1})|, \quad (6)$$

where Z is the normalization constant. Compared to some tracking techniques that only generate one state at time k [13], we search multiple predicted states, and assign each a weight (probability). The single-state model is suitable for gradual line changes, while the multiple-state model can predict the correct state even when the line orientation changes abruptly.

3.2 Matching Model

The measurement vector represents observations that are related to the predicted state. The measurement vector contains a textural and an orientation characteristics (see Section 2.1):

$$D_k = \begin{bmatrix} g_k \\ o_k \end{bmatrix}, \quad (7)$$

where g_k and o_k are the textural and orientation values at pixel (x_k, y_k) . The matching model is defined to match the observations at predicted states to the reference profile. According to the linear structure characteristics, continuity and consistency, the texture and orientation of the observation D_k at state X_k should be similar to the reference profile denoted by $D' = [g', o']$. The matching process can be modeled as the sum of two Gaussian functions.

$$P(D_k|X_k) = \frac{1}{Z} \left(\exp\left(-\frac{(g_k - g')^2}{2\sigma_g^2}\right) + \exp\left(-\frac{(o_k - o')^2}{2\sigma_o^2}\right) \right) \quad (8)$$

In order to incorporate dynamic features of linear structures, such as surface or direction changes, the reference

is not kept constant but is updated dynamically. The reference profile $D' = [g', o']$ at time k is determined by the observation measurements from the previous state sequence $X_{k-n:k-1}$,

$$\begin{bmatrix} g' \\ o' \end{bmatrix} = \sum_{i=1}^n P_{k-i} \begin{bmatrix} g_{k-i} \\ o_{k-i} \end{bmatrix}, \quad (9)$$

where P_{k-i} represents the weighting coefficient that is determined by the posterior probability of state X_{k-i} , and n is the number of past observations involved to estimate the reference pixel in each tracking step. Smaller values of n increase sensitivity to changes of the line, but are more likely to include noisy references.

3.3 Tracking Process

Given the state sequence $X_{0:k-1}$ and the observation sequence $D_{0:k-1}$, the tracking process can be modeled as a posterior probability $P(X_k|D_{0:k})$. According to the Bayes rule

$$P(A|B, C) = \frac{P(B|A, C)P(A|C)}{P(B|C)},$$

we obtain the posterior probability

$$\begin{aligned} P(X_k|D_{0:k}) &= P(X_k|D_{0:k-1}, D_k) \\ &= \frac{P(D_k|X_k, D_{0:k-1})P(X_k|D_{0:k-1})}{P(D_k|D_{0:k-1})}. \end{aligned}$$

Since the observation D_k is independent of the previous observation sequence $D_{0:k-1}$, $P(D_k|D_{0:k-1})$ is a normalizing constant, and we obtain

$$P(D_k|X_k, D_{0:k-1}) = P(D_k|X_k), \quad (10)$$

which can be estimated from the matching model given in Equation 8.

Given the posterior probability $P(X_{k-1}|D_{0:k-1})$ of the previous state, the prior probability of the state X_k can be recursively estimated by the Chapman-Kolmogorov equation [1]

$$P(X_k|D_{0:k-1}) = \sum_{k=1}^N P(X_k|X_{k-1})P(X_{k-1}|D_{0:k-1}),$$

where N is the number of possible states at time $k-1$, and $P(X_k|X_{k-1})$ can be estimated by the prediction model Equation 6. Hence, the tracking state X_k at time k can be recursively updated by the maximal posterior probability

$$\begin{aligned} P(X_k|D_{0:k}) &\propto P(D_k|X_k) \times \\ &\sum_{k=1}^N P(X_k|X_{k-1})P(X_{k-1}|D_{0:k-1}) \end{aligned} \quad (11)$$

3.4 Stopping Criteria

One way to stop the tracking process is based on thresholds. Equation 11 shows that tracking performance depends on the predicted state and the correlation between the predicted profile and the reference profile. Tracking should stop if the posterior probability in Equation 11 at time k falls below the threshold.

In our system, stopping is based on the line detection and the image interpretation results. Since image interpretation based on MRF (see Section 2.3) can label almost all the prevalent lines in aerial images, tracking should proceed only in the vicinity of labeled lines. Tracking should stop if it deviates too much from labeled lines. This can occur either when a tracking error occurs or when the end of a line (pipeline or road) has been reached.

3.5 Tracking versus Inference

The identification of connections between linear structures is formulated here as a dynamic tracking problem. The tracker inspects positions sequentially, greedily picking the position with the highest probability. Alternatively, one could reformulate the problem as an inference problem in which one maximizes $P(X_{0:k}|D_{0:k})$. This would make it possible to use other methods, including dynamic programming, active contours [2], level sets [11], and others. In contrast, we use dynamic tracking because the system is imbedded in a semi-automatic system for interpreting aerial images. In this system, the computer acts as an assistant for the human operator, learns simple image interpretations *online*, takes over simple tasks such as tracking roads or pipelines, and returns control to the operator whenever a problem is encountered [14, 15]. It is thus important that the computer proceeds in a sequential manner, permitting the operator to stop the interpretation process at any moment without having to correct or delete too many incorrect interpretations.

4 Experimental Results

The criteria used to evaluate a tracking system are based on accuracy and completeness. Accuracy refers to the number of tracking pixels that are correct, and completeness refers to the number of lines that are tracked completely. Accuracy should be given a higher priority than completeness because - at least in semi-automatic systems [13] - incomplete tracking can be continued after an interaction with the human operator.

Our experiments were performed on the sub-images extracted from 7140×5940 and 7080×5880 aerial photos with a resolution of 5 meters per pixel. The parameters of the prediction model were set to $d=10$ pixels, and

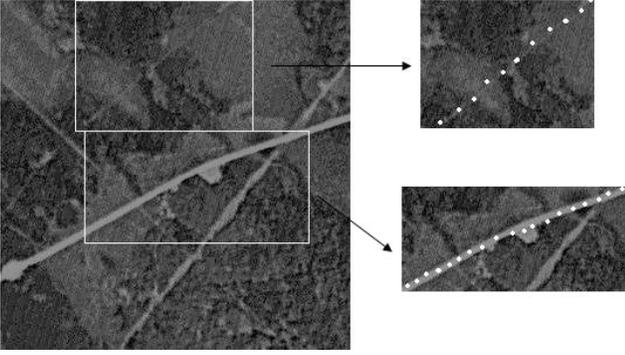


Figure 6. Aerial image and road tracking for two subimages.

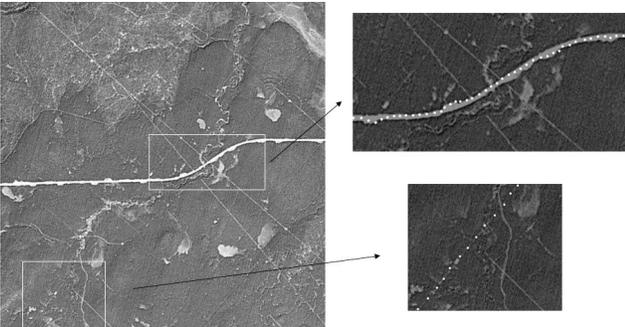


Figure 7. Aerial image and road tracking for two subimages.

$\theta'_k = -\frac{\pi}{6} + \frac{\pi}{18} * i, i = 0, \dots, 6$. The parameters of the matching model were set to $\sigma_g = 0.5$ and $\sigma_o = 2$ with $o = 10$, given that we used a bank of 10 Gabor filters (see Section 2.1).

As described in Section 2, the Bayesian tracker is responsible for two tasks, namely for tracking orientation and textural changes. For texture tracking, it attempts to deal with occlusions such as shown in Figures 6 and 7, which could not be detected in the line detection stage. Orientation tracking involves tracking curved roads and curved connections between segments of linear structures. Road tracking results are shown in Figures 6 and 7. Both images are very large, hence we show tracking details for two subimages. The tracking positions, which are indicated by white dots, are separated by $d=10$ pixels, as indicated in the previous paragraph. Both images indicate that curved roads can be tracked successfully.

Second, the Bayesian tracker is responsible for tracking curved connections between segments of linear structures. The linear structures are detected by the mechanisms described in Sections 2.2 and 2.3. This is illustrated in Fig-

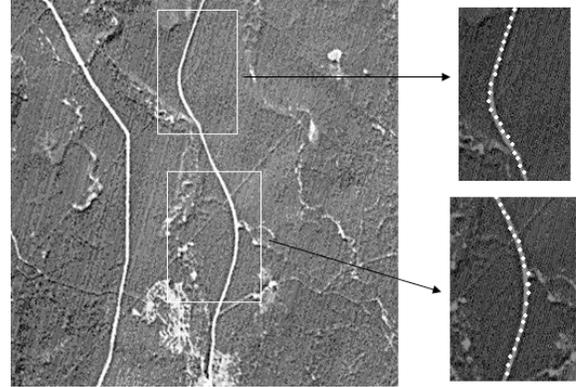


Figure 8. Aerial image with linear structures detected by the mechanisms described in Section 2 and connections found by the Bayesian tracker described in Section 3

ure 4. The Bayesian tracker follows the curved paths connecting the line segments, as illustrated in Figure 8.

Tracking performance depends on the step size d of the prediction model. Tracking with larger values of d is suitable for straight lines and robust to small occlusions; tracking with smaller values of d is more sensitive to changes in line directions, but is also more affected by occlusions and noise. This is illustrated in Figure 9, where tracking is performed with a step size $d = 10$ in panels a and c, and with a step size $d = 30$ in panels b and d. Using a larger value of d leads to decreased accuracy in tracking the curved road (Panel b); using a smaller value of d leads to decreased accuracy in tracking due to occlusions and noise (Panel c).

5 Conclusions

We presented a Bayesian model of line tracking. It attempted to deal with occlusions and curve conjunctions of the linear structures in aerial images. Bayesian tracking was performed in three stages, prediction of the possible states, matching the profile with the reference, and updating the tracking state. The tracking process can be initialized automatically without manual input. The experimental results show that our proposed method could describe the dynamic changes of line direction or texture and that it was robust to the small occlusions in low resolution aerial images.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

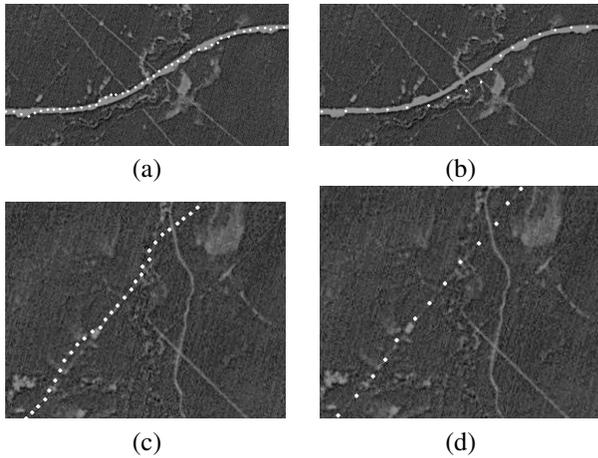


Figure 9. Effect of step size on tracking performance.

References

- [1] S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [2] A. Blake and M. Isard. *Active Contours*. Springer, 1999.
- [3] G. Bordes, G. Giraudon, and O. Jamet. Road modeling based on a cartographic database for aerial image interpretation. In *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, pages 123–139. Birkhauser Verlag, 1997.
- [4] S. A. Dudani and A. L. Luk. Locating straight-line edge segments on outdoor scenes. *Pattern Recognition*, 10:145–157, 1978.
- [5] R. Gao and W. F. Bischof. Detection of linear features in aerial images. Master’s thesis, University of Alberta, 2009.
- [6] R. Gao and W. F. Bischof. Detection of linear structures in remote-sensed images. International Conference on Image Analysis and Recognition, (in submission), 2009.
- [7] A. Grün and H. Li. Linear feature extraction with 3D LSB-snakes. In A. Grün, O. Kübler, and P. Agouris, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 287–298, 1997.
- [8] L. Kennedy and M. Basu. Image enhancement using a human visual system model. *Pattern Recognition*, 30(12):2001–2014, 1997.
- [9] R. Manthalkar, P. Biswas, and B. Chatterji. Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognition Letters*, 24:2061–2068, 2003.
- [10] D. McKeown and J. Denlinger. Cooperative methods for road tracking in aerial imagery. In *Proceedings of International Conference on Compute Vision and Pattern Recognition*, pages 662–672, 1988.
- [11] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [12] G. Vosselman and J. Knecht. Road tracing by profile matching and Kalman filtering. In *Proceedings of the Workshop of Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 265–272, 1995.
- [13] J. Zhou, W. Bischof, and T. Caelli. Road tracking in aerial images based on human-computer interaction and Bayesian filtering. *Photogrammetry and Remote Sensing*, 6(2):108–124, 2006.
- [14] J. Zhou, L. Cheng, and W. F. Bischof. Online learning with novelty detection in human-guided road tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 45:3967–3977, 2007.
- [15] J. Zhou, L. Cheng, and W. F. Bischof. Prediction and change detection in sequential data for interactive applications. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 805–810, 2008.