ROBUST AND EFFICIENT ROAD TRACKING IN AERIAL IMAGES

J. Zhou[†*], W. F. Bischof[†], T. Caelli[‡]

[†]Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada
(jzhou,wfb)@cs.ualberta.ca
[‡]National ICT Australia, Canberra Laboratory, Locked Bag 8001, Canberra ACT 2601, Australia
Terry.Caelli@nicta.com.au

Commission III, WG III/5

KEY WORDS: map revision, road tracking, particle filter, human-computer interaction

ABSTRACT:

Automated road tracking is important for map revision but is currently not reliable enough to be useful for industrial applications. Consequently semi-automatic road tracking has become the preferred solution. In this paper we introduce a road tracking system based on particle filtering and human-computer interactions. Particle filters were used to estimate road axis points. During the estimation and human-computer interaction, new reference profiles were generated and stored in the road template memory for future correlation analysis, thus covering the space of road profiles. Human input provided the road tracker with initial estimates, updated state parameters and multiple reference profiles. This approach has resulted in remarkable improvements in efficiency, compared to the human-only approach while preserving robustness and accuracy.

1 INTRODUCTION

Map revision is time consuming and expensive. It has been reported that, for a number of reasons (Groat, 2003), the average age of topographic maps from United States Geological survey (USGS) is more than 23 years. To solve this problem, many researchers in Computer Vision, Pattern Recognition and Remote Sensing have made efforts to automate the process (Mena, 2003; Rosenfeld, 2000). Road extraction methods are typically automatic or semi-automatic, depending on whether a human operator is involved in the process. Automatic methods aim at replacing humans (Mckeown et al., 1998; Geman and Jedynak, 1996; Klang, 1998). However, humans cannot be excluded completely from the revision process in real applications because the computer vision algorithm are not sufficiently robust and reliable and, importantly, a map is a legal document requiring some form of final checking by a human. For these reasons, semi-automatic methods are preferable (Laptev et al., 2000).

Mckeown and Denlinger (1988) introduced a semi-automatic road tracker based on road profile correlation and road edge following. The tracker was initialized by a human operator to obtain starting values for coordinates, direction and width of the road. Road axis points were then predicted by a road trajectory model and correlation models. The edge-based tracker modelled the road by linking points with high gradient and orientation in the expected direction.

Vosselman and Knecht (Vosselman and Knecht, 1995) proposed a road tracker based on a single observation model Kalman filter. Human input was used to initialize the state of the Kalman filter and to extract a template road profile. The Kalman filter then recursively updated its state to predict the road center, using feedback from matching the template profiles to the observed profiles. Baumgartner et al. (2002) developed a prototype system based on the above method. An interaction interface was designed to coordinate human actions with computer predictions.

In this paper, we present a semi-automatic road tracking system

---

[*]Corresponding author.

based on particle filtering. The particle filters are used to estimate the current state of the system based on past and current observations. When the particle filter fails, the human operator initializes another filter from the failure point. Observation profiles are generated from 2-D features of the road texture, making the tracker more robust. The optimal profile match is determined from the current state of the particle filter and the multiple observations. The human operator interacts with the road tracker throughout the tracking process. The operator input not only sets the initial state of the particle filters, but also reflects knowledge on road profiles. Consequently, the road tracker is more intelligent in dealing with different kinds of road situations, including obstructions such as vehicles, bridges, road surfaces changes, and more.

2 SYSTEM OVERVIEW

2.1 Application Background

One of the main paper products of the USGS topographic maps for the USA is 7.5-minute quadrangle topographic map (Groat, 2003). This map series consists of about 55,000 map sheets. The revision of this map series is the Raster Graph Revision (RGR) program, which uses existing map sheets as the primary input and creates new maps as the primary output. The maps are displayed on a computer screen, together with the digital orthophoto quads (DOQs) of the area to be mapped (see Figure 1). DOQs are orthogonally rectified images produced from aerial photos taken at a height of 20, 000 feet, with an approximate scale of 1 : 40, 000, and having a ground resolution of 1 meter. The cartographer then makes a visual comparison of the map and the DOQs. When a discrepancy is found between a feature on the map and the DOQ, the cartographer modifies the map to match the DOQ. This process is tedious and time consuming.

2.2 Prototype Road Tracking System

Road revision is the main task of the RGR program because roads are the most frequently changed map elements. In automating

Figure 1: Map revision environment. Here previous map layers are aligned with current digital image data.

this process, most road tracking methods make the following assumptions about road characteristics (Vosselman and Knecht, 1995):

- roads are elongated,

- road surfaces are usually homogeneous,

- there is adequate contrast between road and adjacent areas.

However, these assumptions are not always true. In curved areas or ramps, the road may not be elongated. The road surface may be built of various materials that appear quite different in the images. Background objects such as trees, houses, vehicles and shadows may occlude the road surface and may strongly influence the road appearance. Road surfaces may not have adequate contrast with the adjacent areas because of road texture, lighting, and weather conditions. Furthermore, the resolution of the aerial images has a significant impact on the computer vision algorithms. Figure 2 shows a typical example of an image extracted from DOQ to provide information for map revision purposes.

Figure 2: An image sample extracted from DOQ: 663 by 423 pixels.

The architecture of the proposed system is shown in Figure 3. The system is composed of preprocessing and tracking modules. Human and machine interact during the production process. The tracking results and the reference profiles extracted from human input are stored so that the computer-based tracking module can access them when necessary.

3 PREPROCESSING

The preprocessing module consists of three components, image smoothing, road width estimation, and initial reference-profile
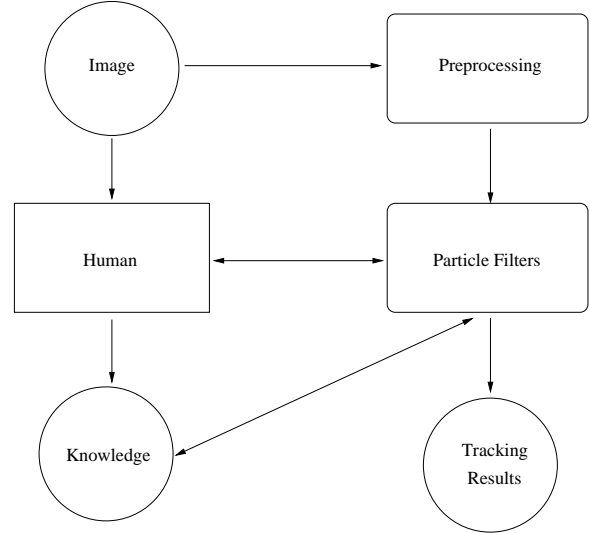

Figure 3: Prototype of the proposed system.

extraction. In the smoothing step, the input image is convolved with a 5×5 Gaussian filter

$$G = \exp(-\frac{x^2 + y^2}{2\sigma^2}) \tag{1}$$

where $\sigma^2 = 2$. This filter was used to set the analysis scale and to reduce high frequency noise.

3.1 Road Width Estimation

Road width determines whether road profiles can be correctly extracted or not. In previous semi-automatic road trackers, the road width was typically entered by the human operator (Mckeown and Denlinger, 1988; Vosselman and Knecht, 1995; Baumgartner et al., 2002). In our system, the road width is estimated automatically, and the human operator only focuses on road axis points and road directions. This is consistent with the operations of road revision in RGR systems. A road segment is entered by the human operator with two consecutive mouse clicks, with the axis joining the points defining the road center line. We assume that the roadsides are straight and parallel lines on both sides of the road axis. Road width can be estimated by calculating the distance between the roadsides. Further, knowledge about road characteristics also helps determining road edges because road width varies as a function of road class.

To detect the road edges, we developed a method based on gradient profiles. This edge detector first estimates the true upper and lower bound of the road width, with the USGS road width definitions serving as a reference (USGS, 1996). At each axis point, a profile is extracted perpendicular to the axis. The length of the profile is bounded by the road width limits defined by USGS. The gradient of the profile is calculated, and one point is selected on both sides of the axis point where the maximum gradient is found. The distance between the two points is considered as the road width at this axis point. For a road axis segment with n points, we obtain a function $f(x)$

$$f(x_i) = number\ of\ times\ x_i\ appears, \quad 1 \leq i \leq n \tag{2}$$

where x_i is the road width value extracted above. n depends on the road width limit from USGS and the complexity of the road conditions. Because the image resolution is 1 meter, x_i correspond to road width of x_i meters. Searching for an x^* where

$$f(x^*) = \arg\max_x f(x_i) \qquad 1 \leq i \leq n \tag{3}$$

yields a dominant road width that appears most of the time. Then new road bounds are calculated using the functions

$$lb = x^* - e \ \ and \ \ ub = x^* + e \tag{4}$$

where lb is the new lower bound, ub is the new upper bound, and $e = 4$ is an empirical value that is proved to be suitable for this application through experiments.

Using the new bounds, the edge detector determines the new road width at each axis point and computes the average as the final road width for profile matching. Figure 4(c) shows how the gradient profile based method generated the edges when the Canny edge detector failed (Canny, 1986).
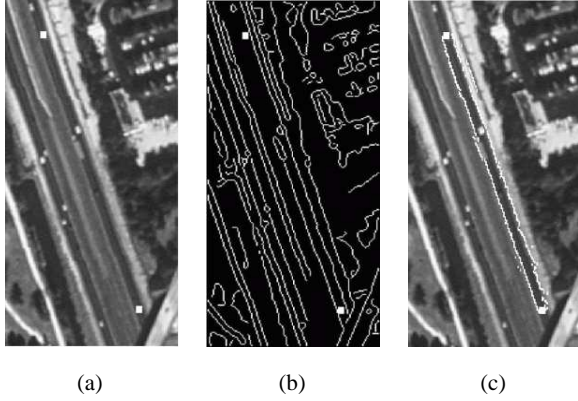

(a)   (b)   (c)

Figure 4: Road edge detection results. (a) Cropped image from DOQ with human input (white blocks). (b) Result of Canny edge detector - notice the multiple road edges. (c) Result of gradient profile based detector - showing just one pair of road edges.

3.2 Reference Profile Extraction

An initial reference profile is extracted from the road segment entered by the human operator. Later, new profiles are extracted and placed into a profile list for further use. Like the extraction of gradient profile in edge detection, a reference profile is extracted at each axis point. Thus, we get a profile sequence that contains the road surface texture information, which may include occluding objects.

For a sequence of road profiles $P = [p_1 p_2 ... p_n]$, profile extraction proceeds as follows. First, an average profile is calculated. Then each profile in the sequence is cross correlated with the average profile. Whenever the correlation coefficient is below a threshold, the profile is removed from the sequence. The algorithm iterates through all the profiles until a new profile sequence is generated. The average profile of the new sequence is taken as the final road segment profile.

Algorithm: REFPROFILE($P = [p_1 p_2 ... p_n]$)

$\bar{p} \leftarrow \frac{\sum_{i=1}^{n} p_i}{n}$
for each p_i
   $r(p_i, \bar{p}) \leftarrow$ correlation coefficient of p_i and $\bar{p}$
   if $r(p_i, \bar{p})$ is smaller than a threshold
   $P \leftarrow P - p_i$
   end **if**
end **for**
$\bar{p} = \frac{\sum_{i=1}^{m} p_i}{m}$

4 ROAD TRACKING

The state evolution of the tracking process can be modelled by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \quad k \in \mathbb{N} \tag{5}$$

where $\mathbf{x}_k$ is the state at time k, $\mathbf{v}_k$ is the process noise, and $\mathbf{f}_k$ is a function of $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$.

Given an observation sequence $\mathbf{z}_{1:k}$, the tracker recursively estimates $\mathbf{x}_k$ the prior probability density function $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the posterior probability density function $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. The relationship between observations and states is defined by

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \quad k \in \mathbb{N} \tag{6}$$

where $\mathbf{n}_k$ is the measurement noise.

Depending on the properties of the state evolution and the posterior density, the tracking problem can be solved with different approaches, such as Kalman filters, hidden Markov models, extended Kalman filters and particle filters (Kalman, 1960; Arulampalam et al., 2002).

4.1 State Model

In our road tracking application, we want to track the road axis points using recursive estimation. Vosselman and Knecht (Vosselman and Knecht, 1995) proposed the following state model:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ \theta' \end{bmatrix} \tag{7}$$

where x and y are the coordinates of road axis points, θ is the direction of the road, and θ' is the change in road direction. The state model is updated by the following non-linear function

$$\mathbf{x}_k = \begin{bmatrix} x_{k-1} + \tau \cos(\theta_{k-1} + \tau \frac{\theta'_{k-1}}{2}) \\ y_{k-1} + \tau \sin(\theta_{k-1} + \tau \frac{\theta'_{k-1}}{2}) \\ \theta_{k-1} + \tau \theta'_{k-1} \\ \theta'_{k-1} \end{bmatrix} \tag{8}$$

where τ is the interval between time $k-1$ and k. The deviation of this simplified dynamics and the true road shape are interpreted as the process noise $\mathbf{v}_k$, whose covariance matrix is Q_k.

4.2 Observation Model

Observations are obtained by matching the reference profiles and the searching profiles. Observation profiles are extracted perpendicular to the road direction at the position estimated by the state models. To minimize the disturbance from background objects on the road and the road surfaces changes, a heuristic search method is used to search the neighborhood of the estimated points for better matching. Euclidean distances between the matching and searching profiles are calculated. The position with the minimum distance is selected as the optimal observation in this iteration.

In this way, the observation $\mathbf{z}_k$ can be calculated as

$$\mathbf{z}_k = \begin{bmatrix} x_k - s_k \sin(\theta_k + \alpha_k) \\ y_k + s_k \cos(\theta_k + \alpha_k) \end{bmatrix} \tag{9}$$

where s_k is a shift from the estimated road axis point and α_k is a small change to the estimated road direction.

4.3 Particle Filtering

Particle filtering, specifically the CONDENSATION algorithm proposed by Isard and Blake (1998), has been successfully used in modelling non-linear and non-Gaussian processes (Arulampalam et al., 2002; Southall and Taylor, 2001; Lee et al., 2002). The filter approximates the posterior density $p(\mathbf{x}_k|\mathbf{z}_k)$ by the sample set $\{\mathbf{s}_k^i, \pi_k^i, \quad i = 1, ..., N\}$ at each time step k, where π_k^i is a weight to characterize the probability of the particle $\mathbf{s}_k^i$.

Given the sample set $\{\mathbf{s}_{k-1}^i, \pi_{k-1}^i, \quad i = 1, ..., N\}$ at time $k-1$, the iteration of the particle filter can be summarized as follows:

1. Construct cumulative density functions $\{c_{k-1}^i\}$ for each sample from the current sample set. Randomly select N samples $\{\mathbf{x}_{k-1}^j, \quad j = 1, ..., N\}$ according to the density function.

2. Update each sample by equation 5 to generate new samples $\{\mathbf{x}_k^j, \quad j = 1, ..., N\}$

3. Calculate new weights for each sample based on observation $\mathbf{z}_k$. The weights are normalized and are proportional to the likelihood $p(\mathbf{z}_k|\mathbf{x}_k^j)$. In this way, a new sample set $\{\mathbf{s}_k^i, \pi_k^i, \quad i = 1, ..., N\}$ is constructed.

Then the estimated state at time k is

$$E(\mathbf{x}_k) = \sum_{i=1}^{N} \pi_k^i \mathbf{s}_k^i \qquad (10)$$

In our application, we assume that the observation is normally distributed with standard deviation σ, thus the likelihood for the observation has the property

$$p(\mathbf{z}|\mathbf{x}^j) \propto \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{d_j^2}{2\sigma^2}) \qquad (11)$$

where d_j is the Euclidean distance between the position of particle $\mathbf{x}^j$ and the observation.

The number of particles is set to 10 times the number of pixels correspond to road width. The initial density of $p(\mathbf{x}_0)$ is set to be uniformly distributed. The particle filter gradually adjusts the weights of each particle during the evolution process.

4.4 Stopping Criteria

A matching profile can be extracted from the observation model. This profile is cross-correlated with the reference profile. If the correlation coefficient exceeds some threshold (e.g. 0.8 in (Vosselman and Knecht, 1995)), then the observation is accepted. If the coefficient is below the threshold, and some other conditions are met (e.g. a high contrast between the profiles), the observation is rejected. In this case, the particle filter makes another state update based on the current state, and using a larger time interval τ, so that the estimated position is jumped over. When multiple jumps occur, the particle filter recognizes it as a tracking failure and returns control back to the human operator (Baumgartner et al., 2002).

However, road characteristics are more complex in real applications. Cross-correlation may not always generate a reliable profile match, which leads to errors in the tracking process. Furthermore, the particle filter may fail frequently because the predicted position may not contain an observation profile that matches the reference profile. The system then requires substantial interactions with the human operator, making the tracking process less efficient and quite annoying.

4.5 Improving Robustness

To improve robustness, we carefully selected the features used in profile extraction. We used two-dimensional road features in observation and reference profiles. In addition to searching along a line perpendicular to the road direction, we also search a line along the road direction. Profiles are extracted in both directions and combined. The parallel profile is useful because greylevels vary very little along the road direction, whereas this is not the case in off-road areas. Thus the risk of off-road tracking is reduced, and, in turn, tracking errors are reduced.

4.6 Improving Efficiency

In previous methods (Vosselman and Knecht, 1995; Baumgartner et al., 2002), each time a new reference profile was extracted, the old reference profile was discarded. In our system, all the reference profiles are retained. Thus, the road tracker gradually accumulates knowledge on road conditions. During profile matching, the latest profile is given the highest priority. When matching fails, the road tracker searches the reference profile list for a match. To reflect the gradual change of the road texture, the reference profile is updated by successful matches, using a weighted sum.

We developed an algorithm to search for the optimal observation-reference profile combination. The search space $V = \langle X, Y, \Theta \rangle$ is defined by the current state $\mathbf{x}$, where X, Y and Θ are bounded by a small neighborhood of x, y and θ respectively. The searching algorithm is described below:

Algorithm: OPTPROFILE($P = p_1, p_2, ..., p_n, V$)

    for each $v_i \in V$
        extract profile p_i' at v_i
        $c(p_i', p_1)$ ←cross correlation coefficient of p_i' and p_1
    end for
    $c^* = max(c(p_i', p_1))$
    if $c^* > 0.9$
        update p_1
        return v^*
    else
        for each p_i'
            for each $p_j \in P, j \neq 1$
                $c(p_i', p_j)$ ←cross correlation coefficient
                      of p_i' and p_j
            end for
        end for
        $c^* = max(c(p_i', p_j))$
        if $c^* > 0.9$
            $p^* = p_j$ correspond to c^*
            switch p_1 and p^*
            return v^*
        else
            return rejection
        end if
    end if

Image multi-scaling is known to be useful for improving the efficiency of road tracking. Indeed, humans use multiscale attention in many tasks to focus on important features and reduce distractors(LaBerge, 1995). To simulate such behaviour, we added a scale parameter to the state update model of the particle filters. After a series of successful matches, the scale parameter is increased, the step size is increased, and hence the speed of the tracking process in increased.

5 EXPERIMENTS AND EVALUATIONS

24 images were extracted from DOQs of Marietta, Florida. The images included scenes of trans-national highways, intra-state highways and roads for local transportation. Further, they contained different road types, such as straight roads, curves, ramps, crossings, and bridges. The images also contain various road conditions, including occlusions by vehicles, trees, and shadows.

A trained human operator was required to draw roads by hand in a real map revision environment as used at USGS. The drawing was performed by selecting tools for specific road classes, followed by mouse clicks on the road axis point in the images. Both spatial and temporal information of the human input were recorded, and only inputs on road tracking were kept. This data was used to initialize the particle filters, to regain control when the road tracker had failed, and it was also used to compare performance with the road tracker. A total of 342 human inputs were recorded with a total time of 1167 seconds. Hence each image took on average 48.6 seconds and 14.3 inputs.

Tracking performance was evaluated in three respects, correctness, savings in human input, and savings in plotting time. Correctness has the highest priority because if an error happens, the human operator has to remove the error segment and then draw a new one. This may take longer than the time that was initially saved. The number of human inputs and plotting time are related, hence reducing the number of human inputs also decreases plotting time. Based on the average time for a human input, we get an empirical function to calculate the time cost of the road tracker:

$$t_c = \begin{cases} t_h & if\ an\ error\ is\ present, \\ t_t + 4n_h & if\ no\ error\ is\ present. \end{cases} \quad (12)$$

where t_c is the total time cost, t_h is the time cost if plotting by human, t_t is the time used in tracking by the road tracker, and n_h is the number of human inputs required during the tracking.

Table 1 shows the performance comparison between our road trackers and a human operator working without the system. The proposed road tracking system demonstrates substantial cost savings in the number of human inputs and the time. On average, human inputs and time cost are reduced by 73.4% and 57.4% using the tracker based on particle filtering.

	human	PF
tracking errors	0	2
time cost	1167	497
human inputs	342	91

Table 1: Road tracking results

In the road tracking application, the system states and the observations are subject to noise from different sources, including those caused by image generation, disturbances on the road surface, road curvature changes, as well as other unknown sources. The state evolution process propagates the noise into the state pdf. It is thus better to construct a non-Gaussian and multi-modal pdf. The particle filter is a good solution to such problems because it is an approach to modeling non-Gaussian processes. Figure 5 shows one instance of the tracking accuracy of the particle filters, when evaluated by the deviation of the estimated road axis points from those detected by the human.

Some tracking results are shown in Figures 6 to 8. In Figure 6, the road tracking starts from the upper left corner, where the first two white dots show the positions of the human inputs. The following
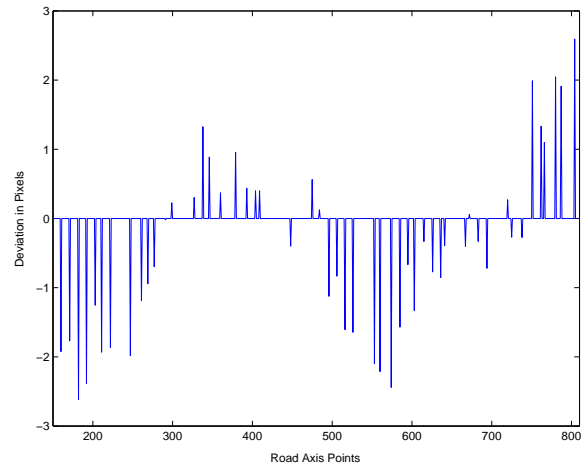

Figure 5: An instance of the tracking accuracy of the particle filter. The graphs show the deviation of the estimated road axis points from those detected by a human operator.

white dots are the road axis points detected by the road tracker. When the texture of the road surface changes, the road tracker failed to predict the next position. The control was given back to the human operator who entered another road segment as marked by a short line segment. Multiple scale operations enable the road tracker to work faster. This can be seen in the image, where larger step sizes are used when consecutive predictions were successful.

Figure 7 shows how multiple reference profiles help the tracking. When the road changes from white to black, a match cannot be found between the observation profile and the current reference profile for the white road. Thus, human input is required. The reference profile extracted from this human input for the black road becomes the current reference profile. When the road changes from black to white, the road tracker searches the whole list of reference profiles for an optimal match because the reference profile for white road is already in the reference profile list. Thus, no new human input is required.

Figure 8 shows an example of tracking errors with particle filtering. The road texture is very close to the off-road background, with the roadside parallel to the road displaying road-like features. Thus, the tracker could not find the correct road axis.

Figure 6: Road tracking from upper left corner to lower right corner. White dots are the detected road axis points, white line segment shows the location of human input.

6 CONCLUSION

We have presented a system for robust and efficient road tracking from aerial images. This approach has a potentially significant impact on the daily work of map revision. It can greatly reduce human effort in the road revision process while guaranteeing accurate results because the user is never removed from the process.

Figure 7: Road tracking from upper left corner to lower right corner. The number of human inputs is reduced by searching in multiple reference profile list, as described in the text.

Figure 8: Road tracking. An example of tracking errors.

The road tracking method is based on particle filters that use a multiple observation profile matching model. The road tracker first estimates the road width and extracts an initial road profile from the human input using edge detection. Then a particle filter is used to predict road axis points by state update equation and correct the predictions by measurement update equation. During the measurement update process multiple observations are obtained at the predicted position. The tracker evaluates the tracking result using normalized cross-correlation between road profiles at previous and at current positions. When multiple profiles are obtained from human input, the profile with the highest cross-correlation coefficient is searched with the most recently used profile being given the highest priority in the search. From time to time, the tracker fails to find points with a sufficiently high cross-correlation. These points are skipped, and control is returned to the human operator if too many points are skipped.

The use of two-dimensional features, multiple observations, and multiple profiles has greatly improved the robustness of the road tracker. When they were combined with multiple scale methods, tracking efficiency was further increased.

REFERENCES

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T., 2002. A tutorial on particle filter for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transaction on Signal Processing*, 50(2):pp. 174–188.

Baumgartner, A., Hinz, S., and Wiedemann, C., 2002. Efficient methods and interfaces for road tracking. *International Archives of Photogrammetry and Remote Sensing*, 34(3B):pp. 28–31.

Canny, J., 1986. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):pp. 679–698.

Geman, D. and Jedynak, B., 1996. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):pp. 1–14.

Groat, C., 2003. The National Map - a continuing, critical need for the nation. *Photogrammetric Engineering & Remote Sensing*, 69(10).

Isard, M. and Blake, A., 1998. CONDENSATION-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):pp. 5–28.

Kalman, R., 1960. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82(D):pp. 35–45.

Klang, D., 1998. Automatic detection of changes in road databases using satellite imagery. *The International Archives of Photogrammetry and Remote Sensing*, 32(4):pp. 293–298.

LaBerge, D., 1995. Computational and anatomical models of selective attention in object identification. In M. Gazzaniga, ed., *The Cognitive Neurosciences*. Cambridge, MA: Bradford.

Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., and Baumgartner, A., 2000. Automatic extraction of roads from aerial images based on scale-space and snakes. *Machine Vision and Applications*, 12(1):pp. 23–31.

Lee, M., Cohen, I., and Jung, S., 2002. Particle filter with analytical inference for human body tracking. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pp. 159–166. Orlando, Florida.

Mckeown, D., Bullwinkle, G., Cochran, S., Harvey, W., McGlone, C., McMahill, J., Polis, M., and Shufelt, J., 1998. Research in image understanding and automated cartography: 1997-1998. Technical report, School of Computer Science, Carnegie Mellon University.

Mckeown, D. and Denlinger, J., 1988. Cooperative methods for road tracing in aerial imagery. In *Proceedings of the IEEE Conference in Computer Vision and Pattern Recognition*, pp. 662–672. Ann Arbor, MI.

Mena, J., 2003. State of the art on automatic road extraction for GIS update: a novel classification. *Pattern Recognition Letters*, 24(16):pp. 3037–3058.

Rosenfeld, A., 2000. Image analysis and computer vision: 1999. *Computer Vision and Image Understanding*, 78(2):pp. 222–302.

Southall, B. and Taylor, C., 2001. Stochastic road shape estimation. In *Proceedings of the Eighth International Conference On Computer Vision*, volume 1, pp. 205–212. Vancouver, Canada.

USGS, 1996. *Standards for 1:24000-Scale Digital Line Graphs and Quadrangle Maps*. U.S. Geological Survey, U.S. Department of the Interior.

Vosselman, G. and Knecht, J., 1995. Road tracing by profile matching and Kalman filtering. In *Proceedings of the Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pp. 265–274. Birkhaeuser, Germany.