Available online at www.sciencedirect.com

ScienceDirect

ISPRS Journal of Photogrammetry & Remote Sensing 61 (2006) 108 – 124

PHOTOGRAMMETRY & REMOTE SENSING

www.elsevier.com/locate/isprsjprs

Road tracking in aerial images based on human–computer interaction and Bayesian filtering

Jun Zhou [a,*], Walter F. Bischof [a], Terry Caelli [b,c]

[a] *Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2E8*
[b] *Canberra Laboratory, National ICT Australia, Locked Bag 8001, Canberra ACT 2601, Australia*
[c] *Research School of Information Science and Engineering, Australian National University, Building 115, Canberra ACT 0200, Australia*

Received 31 January 2006; received in revised form 11 September 2006; accepted 11 September 2006

Abstract

A typical way to update map road layers is to compare recent aerial images with existing map data, detect new roads and add them as cartographic entities to the road layer. This method cannot be fully automated because computer vision algorithms are still not sufficiently robust and reliable. More importantly, maps require final checking by a human due to the legal implications of errors. In this paper we introduce a road tracking system based on human–computer interactions (HCI) and Bayesian filtering. Bayesian filters, specifically, extended Kalman filters and particle filters, are used in conjunction with human inputs to estimate road axis points and update the tracking algorithms. Experimental results show that this approach is efficient and reliable and that it produces substantial savings over the traditional manual map revision approach. The main contribution of the paper is to propose a general and practical system that optimizes the performance of road tracking when both human and computer resources are involved.
© 2006 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

Keywords: Road tracking; Aerial images; Human–computer interaction; Bayesian filtering; Particle filter; Extended Kalman filter

1. Introduction

Map revision is traditionally a manual task especially when maps are updated on the basis of aerial images and existing map data. This is a time consuming and expensive task. For this reason, maps are typically out of date. For example, it has been reported that, for a number of reasons, the revision lag-time for topographic maps from the United States Geological Survey (USGS) is more than 23 years (Groat, 2003).

Road detection and tracking based on computer vision has been one approach to speeding up this process. It requires knowledge about the road database as well as image-related knowledge (Crevier and Lepage, 1997) including the road context, previous processing results, rules, and constraints (Baltsavias, 1997). Many road tracking methods make assumptions about road characteristics (Wang and Newkirk, 1988; Vosselman and Knecht, 1995; Mayer and Steger, 1998; Katartzis et al., 2001; Bentabet et al., 2003; Zlotnick and Carnine, 1993; Mckeown et al., 1998; Klang, 1998; Tupin et al., 2002), including:

- roads are elongated,
- road surfaces are usually homogeneous,

* Corresponding author. Tel.: +1 780 4926554; fax: +1 780 4921071.
E-mail addresses: jzhou@cs.ualberta.ca (J. Zhou), wfb@ualberta.ca (W.F. Bischof), tcaelli@rsise.anu.edu.au (T. Caelli).

0924-2716/$ - see front matter © 2006 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.isprsjprs.2006.09.002

- there is adequate contrast between road and adjacent areas.

One problem with these systems is that such assumptions are pre-defined and fixed whereas image road features vary considerably. For example:

- roads may not be elongated at crossings, bridges, and ramps,
- road surfaces may be built from various materials that cause radiometric changes,
- ground 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 adjacent areas because of road texture, lighting conditions, and weather conditions,
- the resolution of aerial images can have a significant impact on computer vision algorithms.

Such properties cannot be completely predicted and they constitute the main source of problems with fully automated systems.

One solution to this problem is to adopt a semi-automatic approach that retains the "the human in the loop" where computer vision algorithms are used to assist humans performing these tasks (Myers et al., 2000; Pavlovic et al., 1997). In this approach, dynamic knowledge can be transferred to computers, not only when necessary, but also to guide the computer.

Several semi-automatic road tracking systems have been proposed in the past. McKeown and Denlinger (1988) introduced a semi-automatic road tracker based on road profile correlation and road edge following. The tracker was initialized by the user to obtain starting values for position, direction and width of the road. Road axis points were then predicted by a road trajectory and correlation model. Vosselman and Knecht (1995) proposed a road tracker based on a single-observation 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 axis points 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. More recent semi-automatic approaches include the least squares template matching methods (Gruen and Li, 1997) for road centerline extraction by Hu et al. (2004) and Kim et al. (2004), both requiring seed-point input from humans to generate 2D template. Another semi-automatic system was introduced by Xiong and Sperling (2004), who presented a semi-automatic tool that enabled the user to visually check and correct mistakes of the clustering results in performing road network matching.

These semi-automatic systems only allow humans to initiate the tracking process and/or to perform final editing. This makes the road tracking process difficult to control and leaves the combination of human and computer resources suboptimal. Typically, the tracking process is only guided by the most recent human input.

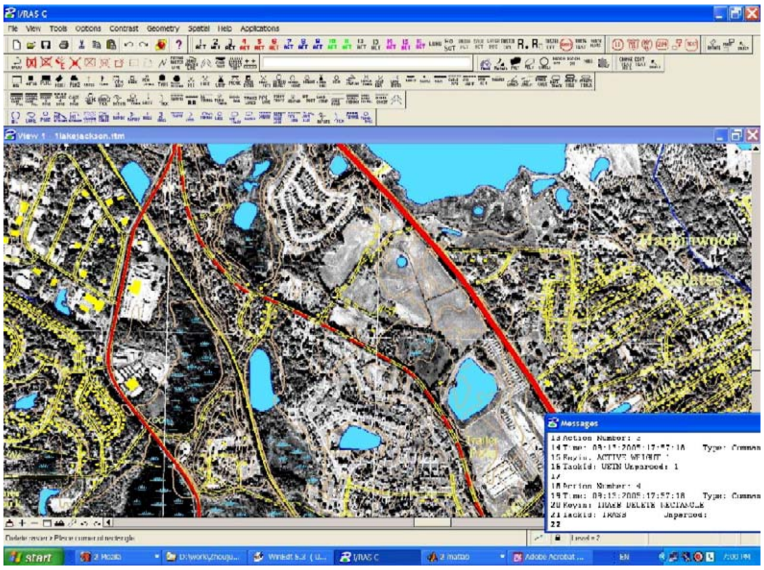
In this paper, we present an approach that uses a semi-automatic road tracking system based on human–computer interaction and Bayesian filtering (Arulampalam et al., 2002). Two models of Bayesian filters, extended Kalman filters and particle filters, are used to estimate the current state of the system based on past and current observations. When the Bayesian filters fail, a human operator observes the reason of the failure and initializes another filter. Observation profiles are generated from 2D features of the road texture making the tracker more robust. Optimal profile matches are determined from the current state of the Bayesian filters and the multiple observations. The human operator interacts with the road tracker not only at the beginning but throughout the tracking process. User input not only sets the initial state of the Bayesian filters but also reflects knowledge of road profiles. Consequently, the road tracker is more flexible in dealing with different kinds of road situations including obstructions by vehicles, bridges, road surfaces changes and more.

The main contribution of the paper is not to develop a new automatic road tracking method, but to propose a general and robust system that effectively combines existing technology with task demands and human performance (Harvey et al., 2004). It is a practical solution to applications in remote sensing and image exploitation, where many automatic algorithms have been developed, but most of them have been unusable in reality (Glatz, 1997).

2. System overview

2.1. Application background

One of the main paper products of the United States Geological Survey (USGS) is the 7.5-minute quadrangle topographic map series (Groat, 2003) which consists of about 55,000 map sheets. The revision of this map series is the Raster Graph Revision (RGR) program which is a tedious and time consuming manual process. The RGR program uses existing map sheets as primary input and


I/RAS C
File View Tools Options Contrast Geometry Spatial Help Applications
View 1 - 1lakejackson.itm
Messages
start

```
13 Action Number: 3
14 Time: 09:13:2005:17:57:18    Type: Comman
15 Keyin: ACTIVE WEIGHT 1
16 TaskId: UEIN Unparsed: 1
17
18 Action Number: 4
19 Time: 09:13:2005:17:57:18    Type: Comman
20 Keyin: ERASE DELETE RECTANGLE
21 TaskId: ERASE        Unparsed:
22
```

Fig. 1. Map revision environment. Previous map layers are aligned with latest digital image data.

creates new maps as primary output. The existing maps are displayed on a computer screen together with the latest digital orthophoto quads (DOQs) of the area to be mapped (see Fig. 1). The cartographer then makes a visual comparison of the map and the DOQs. When an inconsistency is found between a feature on the map and the DOQ the cartographer modifies the map to match the DOQ.

The DOQs are orthogonally rectified images produced from aerial photos taken at height of 20,000 ft, with an approximate scale of 1:40,000 and a ground resolution of 1 m. An example of a road scene in a DOQ is shown in Fig. 2.

2.2. Prototype road tracking system

The road tracking system consists of 5 components (see Fig. 3):

(1) *Human*. The human is the center of control and the source of knowledge.
(2) *Computer*. The computer performs perceptual tasks to replace the human where performance is known to be reliable. It uses vision algorithms for image preprocessing, feature extraction, and tracking.

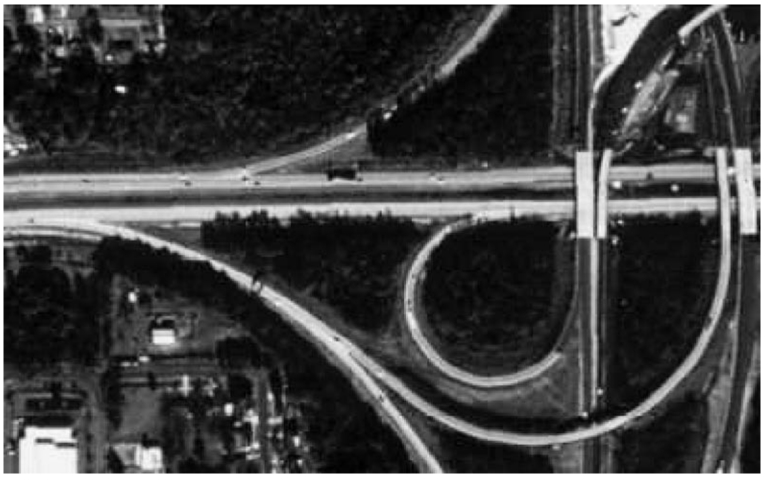


Fig. 2. An image sample of size 663 by 423 pixels extracted from a DOQ.

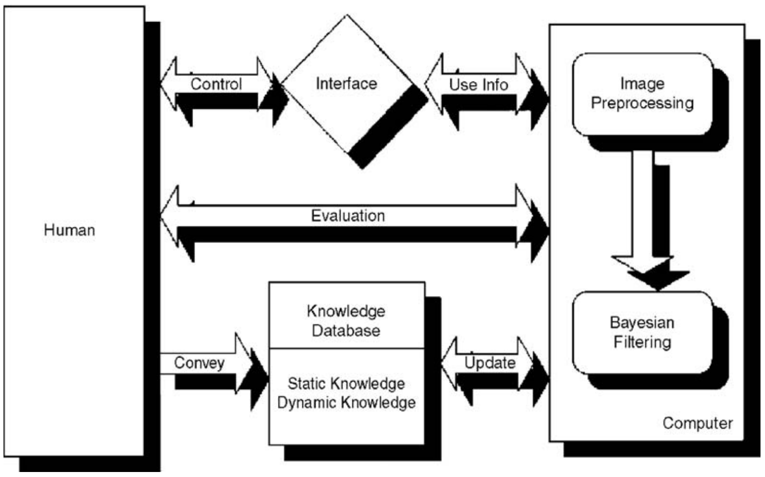


Fig. 3. Block diagram of road tracking system. This system is composed of five components. 1. human; 2. computer vision algorithms to process images and track roads; 3. an interface to track and parse human actions; 4. a database to store knowledge, and 5. evaluation algorithms for feedback purposes, so that the computer can quantitatively evaluate human input and its own performance, and the human can evaluate the performance of computer.

(3) *Human–computer interface*. The choice of the interface has a significant effect on what can be attained. The computer records and parses user's actions in the RGR system through the interface.
(4) *Knowledge transfer scheme.* The computer selects and optimizes vision algorithm parameters based on their ability to predict human behavior throughout the tracking process.
(5) *Performance evaluation.* Performance evaluation is a three-way process in HCI. First, evaluation of human inputs enables the computer to eliminate input noise and decide whether to accept or reject the input. Second, the computer evaluates its own performance and allows the human to gain control over the whole process. Third, the user evaluates the performance of the computer and decides on what to do next.

The road tracking process starts with an initial human input of a road segment, which indicates the road centerline. From this input, the computer learns relevant road information, such as starting location, direction, width, reference profile, and step size. This information is then used to set the initial state model and related parameters of the automatic road tracker. The computer also preprocesses the image to facilitate extraction of road features. The extracted features are compared with knowledge learned from the human operator through profile matching. The computer tracks the road axis points using a Bayesian filter. During tracking, the computer continuously updates road knowledge while, at the same time, evaluating the tracking results. When it detects a possible tracking problem or a tracking failure, it returns control back to the human. The human observes the road changes, diagnoses the failure reason and indicates the correct tracking direction by inputting a new road segment. The new input enables prompt and reliable correction of the state model of the tracker. The new segment can either be input at the current location to resume tracking or can be input at the location where the tracking error happened. In this way errors can be identified and corrected in real time without interrupting the tracking process.

This system architecture enables two knowledge accumulation processes. First, reference profiles extracted from human inputs are stored and the road tracker gradually accumulates knowledge on these reference profiles. These profiles represent different road situations that the tracker has seen. This knowledge passing process makes the tracker increasingly robust. Second, the computer also accumulates knowledge by itself: During tracking, it continues to update the matched reference profiles with the latest tracking results. This enables the tracker to adapt to gradual road changes so that human inputs can be reduced.

The tracking performance is continuously evaluated. When there is lack of confidence over several consecutive positions, the tracker returns control to the human and waits for the next input. The evaluation of new input is based on cross-correlation, i.e. new profiles are defined in terms of their lack of correlation with past ones. In this way, knowledge redundancy is avoided, and the knowledge base does not expand too quickly, thus keeping tracking performance high.

This intelligent tutor/decision maker and apprentice/assistant architecture provides a useful communication path between the human operator and the computer. The computer can learn quickly from humans, and it works more and more independently as tracking goes on.

2.3. Human–computer interface

To enable adequate human–computer interactions we need to track and understand user actions in a software environment. In the USGS map revision system, a simple drawing operation can be implemented by either clicking a tool icon followed by clicking on maps using the mouse, or by entering a key-in command. Each tool in the tool bar corresponds to one cartographic symbol and may encompass a sequence of key-in commands in the execution. Each key-in is considered an event. Events from both inside or outside the system are processed by an input handler and are sent to an input queue. Then a task ID is assigned to each event.

We implemented embedded software to keep track of the states of the event queue and extract detailed information of each event (see Table 1).

We capture and record the time-stamped system-level event sequence, which contains both inter-action and intra-action information. To group the events into meaningful user actions, we analyze and parse the events using natural language processing methods. These include a semantic lexicon for storing action information and a parser for analyzing syntactic and semantic action information. The event sequence is fed into the parser and is segmented into complete actions. A complete description on the human–computer interface is reported in Zhou et al., 2004.

Besides providing analysis of human performance, the interface provides the operator with tools for evaluating the performance of the road tracker. As introduced in later sections, confidence values are generated from the proposed Bayesian filters that allow the performance of the model to be compared directly to the user performance through the realtime tracker interface.

Table 1
Data structure for system-level event

Event ID	ID of the event
Event name	The key-in command
Event type	Is it a key in, coordinate, or reset?
Event time	The time when the event is captured
Event source	Where does the event come from
x coordinate	*x* coordinate of the mouse clicking
y coordinate	*y* coordinate of the mouse clicking

3. Preprocessing

The preprocessing module consists of three components: image smoothing, road width estimation, and extraction of an initial reference-profile. In the smoothing step, the input image is convolved with a 5×5 Gaussian filter

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

where $\sigma = \sqrt{2}$ pixels. This filter was used to set the analysis scale and to reduce high-frequency noise.

3.1. Road width estimation

3.1.1. Method

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 at the beginning of the tracking (McKeown and Denlinger, 1988; Baumgartner et al., 2002), or was estimated automatically in the profile matching step (Vosselman and Knecht, 1995), whereas in our system, the road width is estimated automatically at the beginning of the tracking. 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, a method based on gradient profiles has been developed. 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 along the profile direction is calculated and one point is selected on both sides of the axis point where the largest maximum gradient is found. If several equal largest local maxima are found, the first two local maxima are used. The distance between the two points is considered as the road width at this axis point. For a road axis segment, we obtain a probability density function $p(x)$

$$p(x_i) = \text{number of times } x_i \text{ appears}, \quad 1 \le i \le n \qquad (2)$$

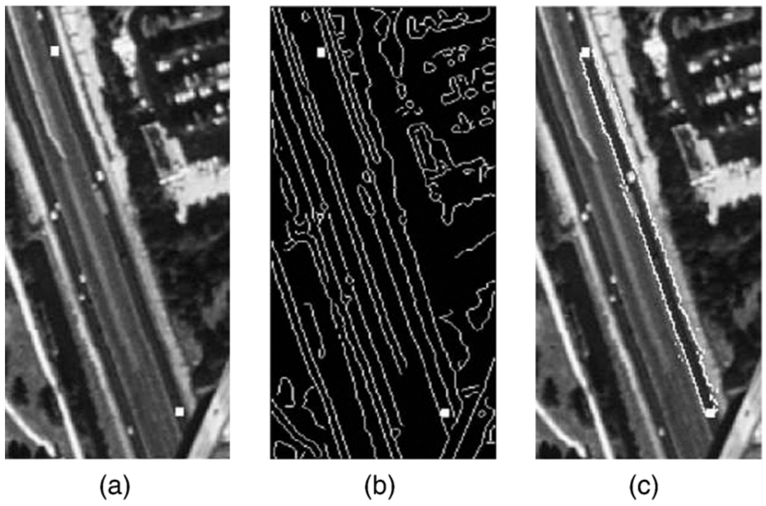


Fig. 4. Road edge detection results. (a) Cropped image from DOQ with human input (white blocks). (b) Result of Canny edge detector: note the presence of multiple road edges. (c) Result of gradient profile based detector: only one pair of road edges is detected.

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 m, x_i corresponds to road width of approximate x_i meters. Searching for an x^* where

$$p(x^*) = \arg \max_x p(x_i) \quad 1 \le i \le n \tag{3}$$

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

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

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

3.1.2. Discussion

Fig. 4 shows road edges detected by the Canny edge detector (Canny, 1986) and our own gradient-based detector. Since the Gaussian filter has already been applied to the image, the implementation of the Canny edge detector starts from calculating the x- and y-gradient. Then the magnitude and direction of the gradient is calculated at each pixel. To perform the non-maximum suppression, we used 0.1 and 0.3 as the low and high threshold to determine the strong edges. Notice that the Canny edge detector does not take advantage of the known road direction and the road width limits, multiple edges may be detected, which causes trouble in finding the true road edges. Thus, our gradient profile based edge detector performs better than the Canny operator, at least in this specific application.

The mean value of the estimated road width was 10.8 pixels, with a standard deviation of 4.3 pixels. In 93.8% of the cases, the estimated road width varied between 6 and 18 pixels, depending on the real road widths, the road conditions, and the locations of human inputs. The estimation of road width can be affected by several factors. First, pavement markings in multi-lane roads, rather than the true road edges, may generate the maximum gradient value. In our application, the aerial images had a resolution of one meter per pixel. Thus, pavement markings were either not wide enough to be displayed or appeared to be less salient after the smoothing step. Second, off-road areas and the road can be made of the same material, or have the same radiometric properties. For example, both the road and the sidewalk could be concrete. In this case, the maximum gradient is not found at the road edge. Our gradient based method either takes the first point at both sides of the road axis as the road edges, or takes the edge of sidewalk as the road edge. In both situations, the road width is bounded by the limits defined by the USGS, so that it does not deviate far from the ground truth.

The reason for estimating the road width automatically was to allow the operator to focus on the road axis points and road directions, consistent with the operation of plotting roads in real-world map revision systems. However, it should be pointed out that tools with manual input are more accurate, though more time consuming, for estimating road widths, as used, for example in the ROADMAP system developed by Harvey et al. (2004).

3.2. Profile extraction

An initial reference profile is extracted as a vector of greylevels from the road segment entered by the human operator. Later, new profiles are extracted from new human inputs and placed into a profile list for further use.

To improve robustness of the system, we use two-dimensional road features, i.e. 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 since greylevel values vary 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. As will be described later in Section 4.2, the observation profiles are also 2D features.

From each human input, we obtain 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 (set to 0.8), the profile is removed from the sequence. In this way, all axis points are evaluated and road profiles extracted from noisy axis points, for example, where cars and trucks are presented, are removed. The algorithm iterates through all the profiles until a new profile sequence is generated, and the average profile of the new sequence is taken as the final road segment profile.

The effectiveness of this noise removal method is affected by road conditions. When occlusions are sparse, the method is quite effective. However, in the case of more populated roads, e.g. roads with a traffic jam, or roads under the shadow of trees, noisy reference profiles may be generated. In these cases, the performance of the system drops.

4. Road tracking

If we consider the road tracking process as a time series, it can be modelled by a state-space approach involving state evolution and noisy measurements. The state evolution of the tracking process can be defined as

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

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

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

$$z_k = h_k(\mathbf{x}_k, n_k) \quad k \in \mathbb{N} \tag{6}$$

where n_k is the measurement noise.

Depending on the properties of the state evolution, the observations, 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; Rabiner, 1989; Welch and Bishop, 1995; Arulampalam et al., 2002). In the following subsections, we introduce two solutions to the tracking problem, extended Kalman filtering and particle filtering.

4.1. State model

Road axis points are tracked using recursive estimation following Vosselman and Knecht (1995), who 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

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

Differences between this simplified process and the true road shape are interpreted as process noise v_k, whose covariance matrix is Q_k.

In Eq. (8), τ is the interval between time $k-1$ and k, determining the distance the road tracker traverses in each step. Initially it is set to the length of the road width and it is affected by three parameters. The first parameter is a “jump-over” factor corresponding to the internal evaluation of the road tracker (see Section 4.5). The second parameter corresponds to the prediction scale (see Section 4.6). The third parameter corresponds to the curvature of the road. When the road curvature is high, a smaller τ is used to avoid off-road tracking.

4.2. Observation model

Observations are obtained by matching the reference profiles to the observed profiles, the latter being extracted in 2D at the position estimated by the state models. To minimize disturbances due to background objects on the road and due to road surfaces changes, a heuristic multiple-observations method is used to search the neighborhood of the estimated points for better matches. Euclidean distances between the matching and observed profiles are calculated, and the position with the minimum distance is selected as the optimal observation in an iteration. The observations z_k are thus calculated as

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

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. Extended Kalman filtering

We have defined a tracking system based on nonlinear state and observation models. Extended Kalman filtering has been widely used to solve such nonlinear time series (Brown and Hwang, 1992; Welch and Bishop, 1995) where the posterior density is assumed to be approximately Gaussian.

The tracking task is performed by estimating the optimal state $\hat{x}$ at each iteration. First, we compute Φ, which contains the coefficients of the linearized time update equations

$$\Phi_k = \frac{\mathrm{d}f_k(x)}{\mathrm{d}x}\Big|_{x=\hat{x}_{k-1}}. \tag{10}$$

The covariance matrix of the predicted state vector becomes

$$P_{k|k-1} = \Phi_k P_{k-1|k-1} \Phi_k^T + Q_{k-1}. \tag{11}$$

After the state update, the extended Kalman filter continues the iteration by solving the following measurement update equations:

$$K_k = P_{k|k-1} A^T (A P_{k|k-1} A^T + R_k)^{-1} \tag{12}$$

$$\hat{x}_k = \Phi_k \hat{x}_{k-1} + K_k (z_k - A \Phi_k \hat{x}_{k-1}) \tag{13}$$

$$P_{k|k} = (I - K_k A) P_{k|k-1} \tag{14}$$

In Eq. (12), A is the measurement matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{15}$$

and R is the covariance matrix of the measurement noise

$$R_k = \sigma^2 \begin{pmatrix} \sin^2(\theta_k) & \sin(\theta_k)\cos(\theta_k) \\ \sin(\theta_k)\cos(\theta_k) & \cos^2(\theta_k) \end{pmatrix}, \tag{16}$$

where σ^2 is the variance of the shift s in the observation model.

The initial state of the Extended Kalman filter is set to $\hat{x}_0 = [x_0\ y_0\ \theta_0\ 0]^T$, where x_0 and y_0 are the coordinates of the end point of the road segment input by human operator, and θ_0 indicates the direction of the road segment. Starting from the initial state, the extended Kalman filter tracks the road axis points iteratively until x or y are outside the image boundaries or a stopping condition has been met.

Vosselman and Knecht (1995) suggested that the covariance matrix Q_k of the process noise in road tracking is mainly determined by the difference between the constant road curvature assumption and the actual curvature changes. They set the standard deviation of the process noise in θ_k' to 1/400 the radius of the road and propagate it to the standard deviation of other state variables. We followed this rule in determining the process noise.

4.4. Particle filtering

Particle filtering, specifically the CONDENSATION algorithm proposed in (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 $p(\mathbf{x}_k|z_k)$ by the particle set $\{s_k^i, w_k^i, i=1,\ldots,N\}$ in each time step k, where w_k^i is a weight used to characterize the probability of the particle s_k^i.

Given the particle set $\{s_{k-1}^i, w_{k-1}^i, i=1,\ldots,N\}$ at time $k-1$, the iteration k of the particle filter can be summarized as follows:

(1) Construct cumulative density functions $\{c_{k-1}^i\}$ on the current particle set. Sample N particles $\{\mathbf{x}_{k-1}^j, j=1,\ldots,N\}$ according to the cumulative density function. The sampling of the jth particle $\mathbf{x}_{k-1}^j$ is done by generating a uniform random

number u^j on [0, 1] and searching for the first particle s^i_{k-1} with $c^i_{k-1} \geq u^j$.

(2) Update each particle by Eq. (8) to generate new particles $\{\mathbf{x}^j_k, j=1,\ldots,N\}$. In the state update, the road curvature parameter θ' is influenced by a zero mean Gaussian random variable with unit variance.

(3) Calculate new weights for each particle based on how well they fit the observation z_k. The weights are normalized and are proportional to the likelihood $p(z_k|\mathbf{x}^j_k)$. In this way, a new particle set $\{s^i_k, w^i_k, i=1,\ldots,N\}$ is constructed.

The estimated state at time k is then

$$E(\mathrm{x}_k) = \sum_{i=1}^{N} w^i_k s^i_k. \tag{17}$$

In our application, we assume that the observation is normally distributed with standard deviation $\sigma = \sqrt{2}$ and so the likelihood of the observation is

$$p\left(z|\mathbf{x}^j\right) \propto \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{d_j^2}{2\sigma^2}\right), \tag{18}$$

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 20 times the road width in pixels. The initial density of $p(\mathbf{x}_0)$ is set to a uniform distribution, which means each particle has the same initial probability. The particle filter gradually adjusts the weights of each particle during the evolution process.

4.5. Stopping criteria

A matching profile is extracted from the observation model and cross-correlated with the reference profile. If the correlation coefficient exceeds some thresholds (e.g. 0.8 in (Vosselman and Knecht, 1995)), 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 Bayesian filters make another state update based on the previous state, using a larger time interval τ, so that the estimated position without accepted observation is jumped over. When contiguous jumps occur (set to 5 jumps), the Bayesian filter recognizes this as a tracking failure and returns control back to the human operator.

The jump-over strategy is particularly useful in dealing with small occlusions on the road, for example, when cars and longtrucks are present. In these cases, the profile matching will not generate high correlation coefficient at the predicted state. The jump over strategy uses an incremented time interval to skip these road positions, so that a state without occlusions can be reached.

In real applications, however, road characteristics are more complex. Cross-correlation may not always generate a meaningful profile match, which in turn may lead to errors in the tracking process. For example, a constant road profile may generate high coefficient when cross-correlated with a profile extracted from an off-road area with constant greylevel. Furthermore, the Bayesian filters may often fail because the predicted position may not contain an observation profile that matches the reference profile. For example, when occlusions are present on the road, the reference and observation profiles may generate a small correlation coefficient, and in turn reject the observation. The system then requires substantial interactions with the human operator, making the tracking process less efficient and quite annoying for the user.

4.6. Improving efficiency

In previous algorithms (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 and the road tracker gradually accumulates knowledge on road conditions. In profile matching, the latest profile is given the highest priority. When matching fails, the Bayesian filters search the list of reference profiles 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 call this the multiple profiles method.

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

```
**Algorithm:** OPTPROFILE($P=p_1, p_2,\ldots, p_n, V$)
  **for** each $v_i \in V$
      extract profile $p'_i$ at $v_i$
      $c(p'_i, p_1) \leftarrow$ 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 $p_j \in P, j \neq 1$
                $c(p'_i, p_j) \leftarrow$ cross-correlation
                        coefficient of $p'_i$ and $p_j$
            end for
    end for
    $c^* = \max(c(p'_i, p_j))$
    if $c^* > 0.9$
        $p^* = p_j$ corresponding to $c^*$
        switch $p_1$ and $p^*$
    return $v^*$
    else
            return rejection
    end if
end if
```

In many tasks, humans use multi-scale attention to focus on important features and to reduce the influence of distractors (LaBerge, 1995). To simulate such behavior, we adopted a step prediction scaling strategy to improve the efficiency of road tracking. A prediction scale is added to the state update model of the Bayesian filters, contributing to the calculation of the time interval τ. The initial prediction scale is set to 1. When a successful match happens the scale parameter is incremented by 1. Whenever matching fails the prediction scale is reset to 1. In this way, the time interval is adjusted automatically. If the road is long, straight and homogenous in surface, the road tracker can predict the next road axis point using a larger scale and ignore many details on the road thus increasing the speed of the tracking process.

5. Experimental results

5.1. Data collection

Eight students were required to plot roads manually in the USGS map revision environment, which displays the old map and the latest DOQ simultaneously on the screen. Plotting was performed by selecting tools for specific road classes, followed by mouse clicks on the perceived road axis points in the image. Before performing the actual annotation, each user was given 30 min to understand the road interpretation process as well as operations such as file input, road plotting, viewing change, and error correction. They did so by working on a real map for the Lake Jackson area in Florida. When they felt confident in using the tools and road recognition, they were assigned 28 tasks to plot roads on the map for the Marietta area in Florida. The users were told that road plotting should be as accurate as possible, i.e. the mouse clicks should be on the true road axis points. Furthermore, the road should be smooth, i.e. abrupt changes in directions should be avoided and no zigzags should occur. Although professional cartographers would be expected to perform such tasks better than the students used here, considering the simplicity of tasks, we believe the performance of students was close to that of experts. Indeed all users became familiar with the annotation operations in less than 15 min.

The plotting tasks included a variety of scenes such as trans-national highways, intra-state highways and roads for local transportation. Further, these tasks contained different road types such as straight roads, curves, ramps, crossings, and bridges. They also included various road conditions such as occlusions by vehicles, trees, or shadows.

Both spatial and temporal information on human inputs were recorded and parsed and only road tracking inputs were kept. We obtained 8 data sets each containing 28 sequences of road axis coordinates tracked by users. Table 2 shows some statistics on the human data. The total time in the table includes the time that users spent on image interpretation, plotting, and error correction. As shown in the evaluation criteria, these were all taken into account in the efficiency calculation.

The system also allowed us to simulate the human–computer interactions using the recorded human data as virtual users. The road trackers interacted with the virtual users throughout the semi-automatic tracking process. Finally, we compared the performance between the simulated semi-automatic road tracking and the complete manual tracking for each user.

5.2. Evaluation

Semi-automatic systems can be evaluated in many ways. For a real-world application, it often includes user

Table 2
Statistics on human input

	User1	User2	User3	User4	User5	User6	User7	User8
Total number of inputs	510	415	419	849	419	583	492	484
Total time (in seconds)	2765	2784	1050	2481	1558	1966	1576	1552
Average number of inputs per task	18.2	15.2	15.0	30.3	15.0	20.8	17.6	17.3
Average time per task (in seconds)	98.8	99.4	37.5	88.6	55.6	70.2	56.3	55.4

Table 3
Comparison of road tracking results between particle filters and extended Kalman filter

	Input saving (%)	Time saving (%)	Distance saving (%)	RMSE (pixels)
Extended Kalman filter	71.9	63.7	85.3	1.86
Particle filter (average)	72.3	62.0	85.6	1.90
Particle filter (best)	79.1	71.6	87.9	2.19

experience evaluation as reported by Harvey et al. (2004). Since our system is still in the simulation stage, we focused on the engineering aspect of the evaluation where the human factors components are only part of the assessment criteria. The criteria used to evaluate this system included the following:

- Correctness: Were there any tracking errors?
- Completeness: Were there any missing road segments?
- Efficiency: How much could tracking save in terms of human input, tracking distance, and plotting time?
- Accuracy: How much did tracking deviate from manual inputs?

Correctness and completeness have the highest priority in Cartography. When errors occur the human operator has to search and correct these errors and this may take longer than the time that was initially saved. The same problem can occur if the update on a road is incomplete. The most important advantage of the proposed system over fully automatic ones is that the human involvement guarantees correctness and completeness of road tracking. The human operator always follows and interacts with the road tracker and whenever an error happens the operator can correct it immediately by initializing a new tracking iteration. The tracking process does not stop until the user decides that all roads have been plotted.

Consequently, in evaluating efficiency, savings in human inputs, in plotting time and in tracking distance have to be considered. The number of human inputs and plotting time are related, so reducing the number of human inputs also decreases plotting time. Given an average time for a human input, which includes the time

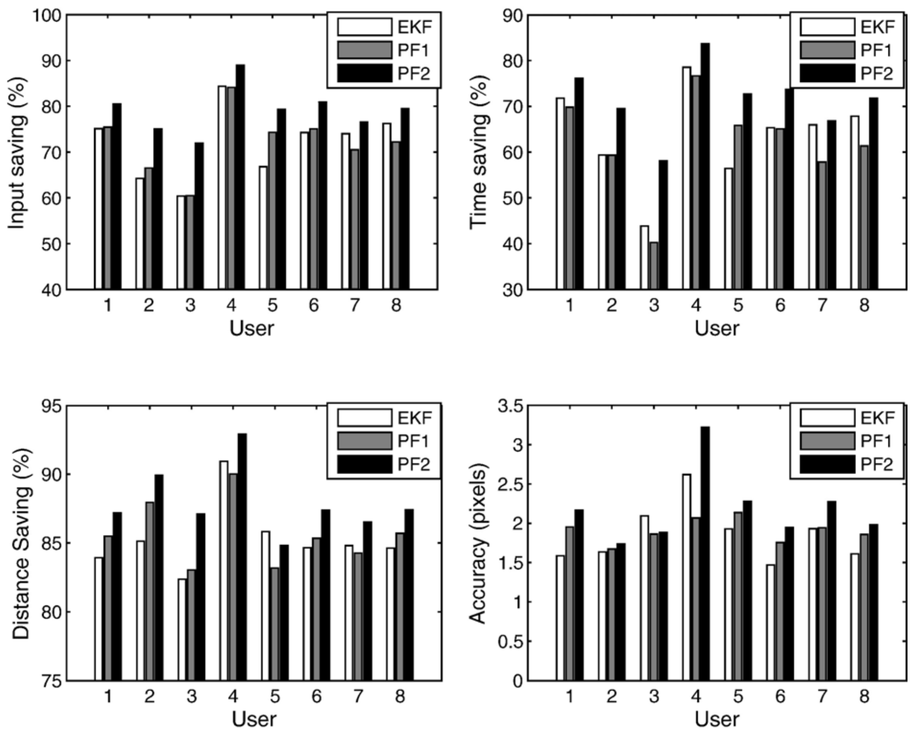


Fig. 5. Comparison of tracking performances for the particle filters (PF1 and PF2) and the extended Kalman filter (EKF). PF1 shows the average performance of the particle filter over 10 Monte Carlo trials. PF2 shows the best performance of the particle filter over 10 Monte Carlo trials. The performance is evaluated on the saving of number of human inputs (upper left graph), the saving of total plotting time (upper right graph), the saving of tracking distance (lower left graph), and the accuracy as the root mean square error of the tracking results against the human input road axis (lower right graph).

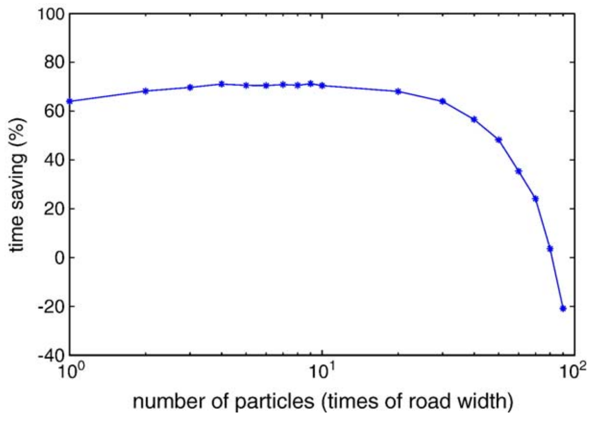


Fig. 6. The influence of number of particles on the performance of the system on data extracted from user one.

for observation, plotting and the switching time between the two, we obtain an empirical function for calculating the time cost of the road tracker:

$$t_c = t_t + \lambda n_h. \tag{19}$$

where t_c is the total time cost, t_t is the tracking time used by road tracker, n_h is the number of human inputs required during the tracking, and λ is an user-specific variable, which is calculated as the average time for an input

$$\lambda_i = \frac{\text{total time for user } i}{\text{total number of inputs for user } i} \quad 1 \le i \le 8 \tag{20}$$

In real application, the λ could be different, depending on the usability of the human–computer interface provided to the user. The savings in tracking distance is defined as the percentage of roads tracked by computer. Tracking accuracy is evaluated as the root mean square error between the road tracker and human input.

5.3. Experimental results

We compared the performance of the particle filter with the extended Kalman filter. Due to the factored sampling involved in the particle filter the tracker may perform differently for each Monte Carlo trial. For this reason we evaluated the particle filter over 10 Monte Carlo trials and we report both average and best performance. Table 3 shows the performance comparison for the road trackers based on particle filtering and extended Kalman filtering. The proposed road tracking system shows substantial efficiency improvement in both non-linear filtering algorithms compared to a human doing the tasks alone. More detailed performance comparison for each user are shown in Fig. 5.

In the proposed road tracking application the system states and observations are subject to noise from different sources including those caused by the image generation, disturbances on the road surface, road curvature changes as well as other unknown sources. The nonlinear state evolution process propagates the noise into the state probability density function (pdf). For this reason it is better to construct a non-Gaussian and multi-modal pdf, and the extended Kalman filter and the particle filter are two sub-optimal methods to solve such systems. The experimental results show that the performance of the extended Kalman filter and the average performance of particle filter are quite similar. Due to the uncertainty in the state evolution of particle filters, different pdf of the states can be approximated in different Monte Carlo trials. When the approximation of

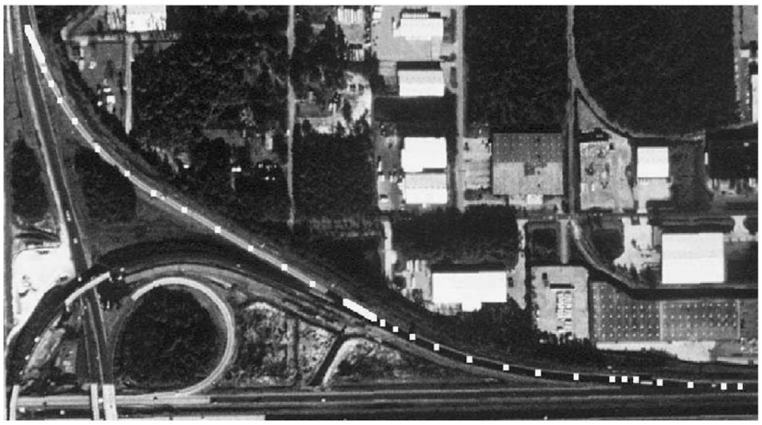


Fig. 7. Road tracking from upper left to lower right. White dots are the detected road axis points, white line segment shows the location of human input.

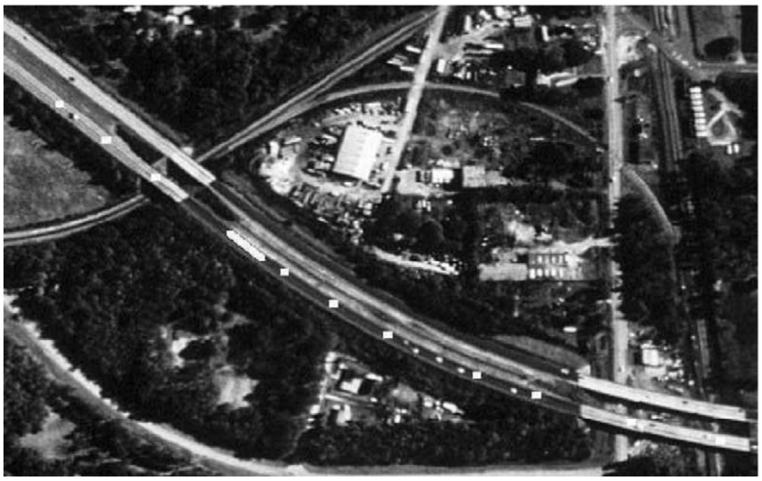


Fig. 8. Road tracking from upper left to lower right. White dots are the detected road axis points, white line segment shows the location of human input. The number of human inputs is reduced by searching multiple reference profile lists, as described in the text.

the state pdf is a better fit to the true pdf, the particle filter out-performs the extended Kalman filter. This is why the best performance of the particle filter is better than that of the extended Kalman filter. We also noticed the compensation on the accuracy in particle filter tracking. This is caused by the error correction function of the particle filters which tolerates more deviations in tracking.

Notice in Table 3 that both Bayesian filters provide approximately the same level of improvement. This suggests that a combination of both filters may further improve the performance of the tracking system. For example, both filters could perform the tracking task simultaneously using a two-filter competing strategy. A dynamic programming approach could also be used to coordinate the behavior of the trackers, using the correlation traces to decide the optimal tracking path. The correlations could also provide human operators with realtime feedback on the ‘‘level of confidence” the system has in the prediction step. This could assist the human operator in monitoring the tracking and in making a decision whether to allow the system to continue tracking.

As can be seen in Fig. 5, the Kalman and particle filters perform differently for different users suggesting that an adaptive, competitive filter selection strategy should be included in the complete tracking model.

Some tracking results are shown in Figs. 7–12. In Fig. 7, the road tracking starts from the upper left corner,

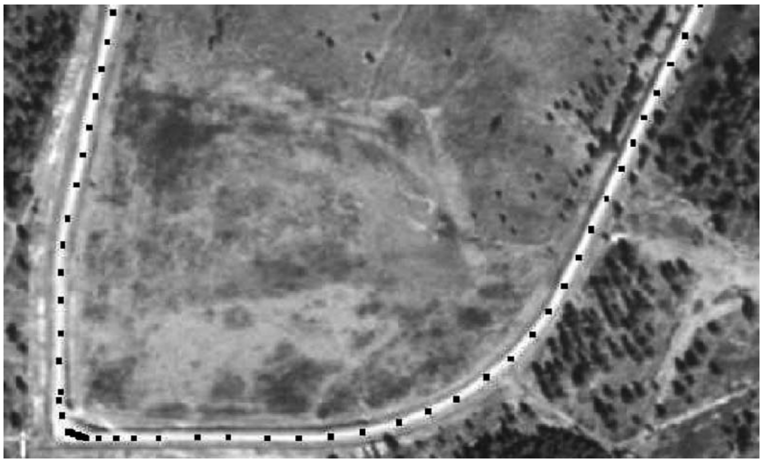


Fig. 9. Road tracking from upper left to upper right. Black dots are the detected road axis points, blackline segment shows the location of human input. The tracking fails when road direction changes dramatically.

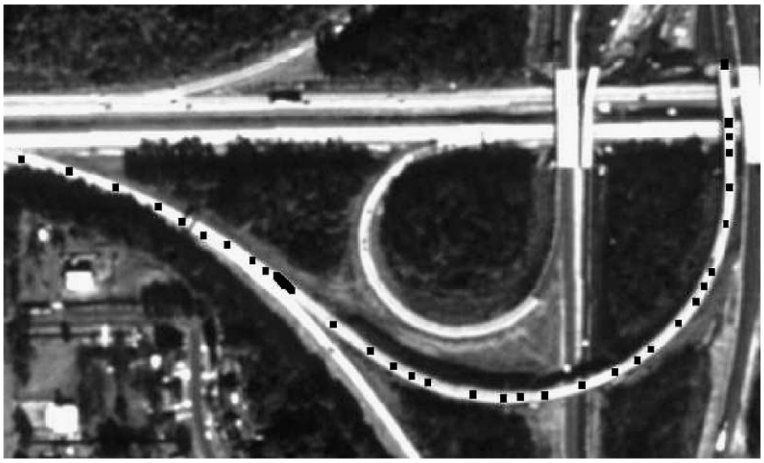


Fig. 10. Road tracking from upper right to upper left. Black dots are the detected road axis points, blackline segment shows the location of human input. The tracking fails when road profile changes at the road connection.

with the white line segment showing the location of human input. The following 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. Control was returned to the user who entered another road segment as marked by a short line segment. Step prediction scaling strategy enables the road tracker to work faster. This can be seen in the image, where larger step sizes are used when consecutive predictions were successful.

Fig. 8 shows how multiple reference profiles help the tracking. Tracking starts at the upper left corner. When the road changes from white to black, a match cannot be found between the observation and the reference profiles and human input is required as indicated by the white line segment. When the road changes back to white, no human input is necessary because the profile for white road is already in the list of reference profiles. The tracker searches the whole list for an optimal match.

The performance of the system is influenced by several factors. First, human factors play an important role. Human input is not always accurate; hence the road tracker is influenced strongly in the preprocessing step when initial parameters are set and road profiles extracted. This is reflected in similar trends of different filtering algorithms tracking in the same data sets. For example, as shown in Fig. 5, the improvement of efficiency of all trackers is the poorest for user 3 and

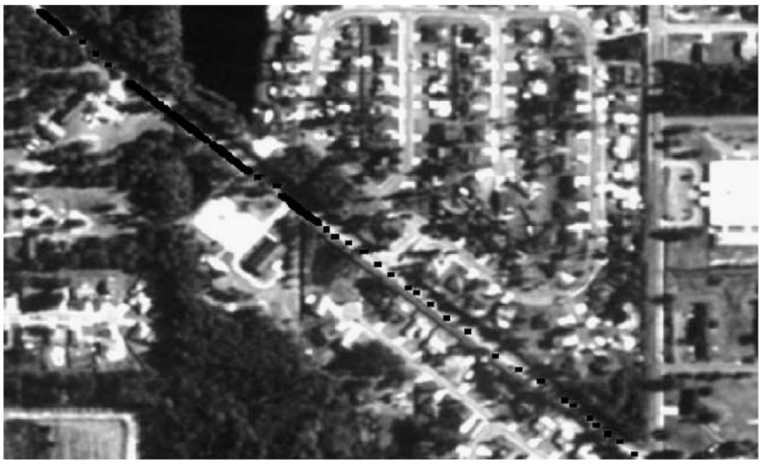


Fig. 11. Road tracking from upper left to lower right. Black dots are the detected road axis points, blackline segment shows the location of human input. This is an extreme case when trees occlude the road. Intensive human inputs are required.

accuracy is the poorest for user 4. This suggests that our system can also be used to model the inputs given by different users. This, in turn, opens the possibility of investigating user-adapted systems.

Second, the number of particles in particle filter affects the performance of the system, as shown in Fig. 6. When the number of particles is smaller than 20 times of the road width, the performance of the system is quite steady. However, when this number continues to increase, the performance of the system drops quickly. Though more particles allow for an improved approximation of the posterior density of the state, the system performance decreases due to the time spent on the particle evolution and likelihood computations.

Third, complex road scene can cause tracking failures, requiring further human input. Figs. 7 and 8 show cases of tracking failures that are caused by abrupt changes of radiometric property of the road. Fig. 9 shows the case that tracking stops where road direction changes abruptly. Figs. 10 and 11 show the tracking failures caused by road profile changes due to road connection and occlusions.

When junctions are encountered the road tracker makes different judgements based on the road condition and the status of the tracker, as shown in Fig. 12. At junction 1, a matching observation could not be found. But further along the direction of the road, a matching observation profile was found. Thus, junction 1 was jumped over by state updating with a large step size τ due to the jump-over strategy (see Section 4.5) or the step prediction scaling strategy (see Section 4.6). However, at junction 2, no matching profile could be found. Thus, the tracking process stopped and control was returned to the human operator. Ultimately, in all difficult situations, it is the human who has to decide how to proceed with tracking.

Vosselman and Knecht (1995) pointed out that a bias may be introduced to the road center estimation, when reference road profiles are updated using successfully matched observation profiles. Unbalanced greylevels of road sides may lead to shift of the road center in the reference road profile. In our system, this is avoided by selecting, in each tracking step, an optimal observation profile from multiple observation profiles, so that the quality of the observation profiles can be improved compared to single observations. The experiments show that updating reference road profile is a compromise. It slightly improves the efficiency of the road tracking, while slightly lowering tracking accuracy. From an HCI point of view, this is a necessary step, as it enables the computer to contribute to knowledge accumulation. We believe this step can be further improved using machine learning approaches.

6. Conclusion

This paper introduced a human–computer interaction system for robust and efficient road tracking. It attempts to bridge the gap between human and computer in automatic or semi-automatic systems. 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. At the same time, it guarantees correct and complete results because the user is never removed from the process.

The proposed framework consists of several components, the user, the human–computer interface, computer vision algorithms, knowledge transfer schemes and evaluation criteria. It can compensate for the deficiencies of computer vision systems in performing tasks usually done by humans. This framework also can be applied to systems that require understanding of different levels of interactions between human and computer.

The road tracking method is based on Bayesian filters that match observation profiles to reference profiles. Particle filters and extended Kalman filters are used to predict road axis points by state update equations and correct the predictions by measurement update equations. 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 the current position. 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. The use of two-dimensional features, multiple observation and multiple profile methods has

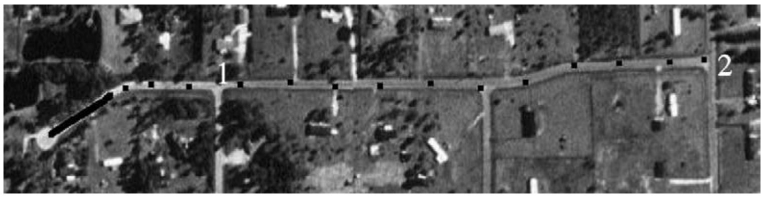


Fig. 12. Handling of road junctions. Blackdots are the detected road axis points. Junction 1 was jumped over, while the tracking stopped at junction 2.

greatly improved the robustness of the road tracker. Finally, when they were combined with step prediction scaling method, tracking efficiency was further increased.

To progress with making human–machine systems more robust and useful we need to explore a number of paths.

- The simulated system approximated the human–computer interaction in a real-world system. To further study the effectiveness and usability of the system, we need to implement it on an industrial platform, such as in the USGS map revision system.
- The combination of Kalman filters and Particle filters should be studied, especially in developing user-adapted systems.
- The system can be more automated and its performance be further improved if more knowledge resources can be involved (for example, the buildings layer).
- The system framework was developed for USGS map revision environment which uses aerial image as the source of revision. It would be interesting to see how this system can be applied to other types of images, for example, to satellite images.

References

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

Baltsavias, E., 1997. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS Journal of Photogrammetry and Remote Sensing 58 (3–4), 129–151.

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

Bentabet, L., Jodouin, S., Ziou, D., Vaillancourt, J., 2003. Road vectors update using SAR imagery: a Snake-based method. IEEE Transaction on Geoscience and Remote Sensing 41 (8), 1785–1803.

Brown, R., Hwang, P., 1992. Introduction to Random Signals and Applied Kalman Filtering, second ed. Wiley.

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

Crevier, D., Lepage, R., 1997. Knowledge-based image understanding systems: a survey. Computer Vision and Image Understanding 67 (2), 161–185.

Glatz, W., 1997. RADIUS and the NEL. In: Firschein, O., Strat, T. (Eds.), RADIUS: Image Understanding for Imagery Intelligence, pp. 3–4.

Groat, C., 2003. The National Map — a continuing, critical need for the nation. Photogrammetric Engineering and Remote Sensing 69 (10), 1087–1090.

Gruen, A., Li, H., 1997. Semi-automatic linear feature extraction by dynamic programming and LSB-snakes. Photogrammetric Engineering and Remote Sensing 63 (8), 985–995.

Harvey, W., McGlone, J., McKeown, D., Irvine, J., 2004. User-centric evaluation of semi-automated road network extraction. Photogrammetric Engineering and Remote Sensing 70 (12), 1353–1364.

Hu, X., Zhang, Z., Tao, C., 2004. A robust method for semi-automatic extraction of road centerlines using a piecewise parabolic model and least square template matching. Photogrammetric Engineering and Remote Sensing 70 (12), 1393–1398.

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

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

Katartzis, A., Sahli, H., Pizurica, V., Cornelis, J., 2001. A model-based approach to the automatic extraction of linear feature from airborne images. IEEE Transaction on Geoscience and Remote Sensing 39 (9), 2073–2079.

Kim, T., Park, S., Kim, M., Jeong, S., Kim, K., 2004. Tracking road centerlines from high resolution remote sensing images by least squares correlation matching. Photogrammetric Engineering and Remote Sensing 70 (12), 1417–1422.

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

LaBerge, D., 1995. Computational and anatomical models of selective attention in object identification. In: Gazzaniga, M. (Ed.), The Cognitive Neurosciences. MIT Press, Cambridge, pp. 649–664.

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

Mayer, H., Steger, C., 1998. Scale-space events and their link to abstraction for road extraction. ISPRS Journal of Photogrammetry and Remote Sensing 53 (2), 62–75.

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

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

Myers, B., Hudson, S.E., Pausch, R., 2000. Past, present, and future of user interface software tools. ACM Transactions on Computer–Human Interaction 7 (1), 3–28.

Pavlovic, V., Sharma, R., Huang, T.S., 1997. Visual interpretation of hand gestures for human–computer interaction: a review. IEEE Transaction on Pattern Analysis and Machine Intelligence 19 (7), 677–695.

Rabiner, L., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77 (2), 257–286.

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

Tupin, F., Houshmand, B., Datcu, F., 2002. Road detection in dense urban areas using SAR imagery and the usefulness of multiple views. IEEE Transaction on Geoscience and Remote Sensing 40 (11), 2405–2414.

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

Wang, F., Newkirk, R., 1988. A knowledge-based system for highway network extraction. IEEE Transactions on Geoscience and Remote Sensing 26 (5), 525–531.

Welch, G., Bishop, G., 1995. An introduction to the Kalman filter. Technical Report. Department of Computer Science, University of North Carolina- Chapel Hill, TR95-041.

Xiong, D., Sperling, J., 2004. Semiautomated matching for network database integration. ISPRS Journal of Photogrammetry and Remote Sensing 59 (1–2), 35–46.

Zhou, J., Bischof, W.F., Caelli, T., 2004. Understanding human–computer interactions in map revision. Proceedings of the 10th International Workshop on Structural and Syntactic Pattern Recognition, Lisbon, Portugal, pp. 287–295.

Zlotnick, A., Carnine, P., 1993. Finding road seeds in aerial images. CVGIP. Image Understanding 57 (2), 243–260.